

AD-A121 899

MODERN CONTROL THEORY FOR ARMY MISSILE GUIDANCE(U)

1/1

INTEGRATED SYSTEMS INC PALO ALTO CA

R H TRAVASSOS ET AL. NOV 82 ISI-21 ARO-18319.1-MA-5

UNCLASSIFIED

DAAG29-81-C-0039

F/G 16/4.2

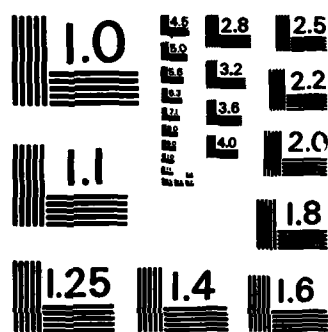
NL

END

FILED

1

Q107



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

12

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 18319.1-MA-S	2. GOVT ACCESSION NO. AD-A221899	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Modern Control Theory for Army Missile Guidance		5. TYPE OF REPORT & PERIOD COVERED Final: 21 Sep 81 - 20 Jun 82
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) R. H. Travassos H. Lev-Ari N. K. Gupta		8. CONTRACT OR GRANT NUMBER(s) DAAG29 81 C 0039
9. PERFORMING ORGANIZATION NAME AND ADDRESS Integrated Systems Incorporated Palo Alto, CA 94301		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Research Office Post Office Box 12211 Research Triangle Park, NC 27709		12. REPORT DATE Nov 82
		13. NUMBER OF PAGES 74
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) N/A		
18. SUPPLEMENTARY NOTES The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) surface to air missiles algorithms guidance systems autopilots trajectories estimating		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) <u>Advances Guidance Laws for Surface-to-Air Missiles:</u> A Singular perturbation guidance law has been developed for medium-range surface-to-air missiles. This guidance law is a significant extension of a previously developed guidance law for short-range missiles; in medium-range intercepts, the problem of energy management should be addressed in addition to homing guidance. The mathematical formulation has been simplified by introducing separation of time scales. <u>Target Trajectory Estimation:</u> A recursive algorithm for estimation of		

DTIC
ELECTE
NOV 30 1982
S D

AD A 121 899

THE FILE COPY

20. ABSTRACT CONTINUED

autoregressive moving average (ARMA) model parameters from noisy samples has been developed. Application of this algorithm to parameter estimation problems has exhibited its fast convergence and unbiasedness in the presence of noise, even with short data records. The algorithm has two versions, a Recursive Maximum Likelihood (RML) form and a Recursive Prediction Error (RPE) form, both of which possess a parallel structure that makes them highly suitable for parallel-processing implementation.

Adaptive Autopilots: Lattice-form algorithms have been developed for fast, recursive identification and control of time-varying systems. These algorithms have excellent numerical properties and a modular structure that makes them suitable for on-board real-time implementation.

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	



Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)



MODERN CONTROL THEORY FOR ARMY MISSILE GUIDANCE

R. H. TRAVASSOS

H. LEV-ARI

N. K. GUPTA

Prepared for:

U. S. Army Research Office

P. O. Box 12211

Research Triangle Park, NC 27709

Attn: Dr. J. Chandra

and

Army Missile Command

Redstone Arsenal, AL 35809

Attn: Dr. W. Kelly

Mr. J. McLean

Prepared under:

Contract DAAG29-81-C-0

ISI 21 • NOVEMBER 1982

**DTIC
ELECTE
NOV 30 1982**

S D

D

PREFACE

The report summarizes research conducted by Integrated Systems, Incorporated, under the Army Research Office Contract No. DAAG29-81-C-0039. The contract was directed by Dr. W. Kelly and Mr. J. McLean at Army Missile Command and by Dr. J. Chandra at the Army Research Office.

At Integrated Systems, Inc., the technical contributions were made by Dr. N. K. Gupta, Dr. H. Lev-Ari, and Dr. R. H. Travassos.

TABLE of CONTENTS

<u>Section</u>		<u>Page</u>
1	INTRODUCTION	1
	1.1 Summary of Approach	2
	1.2 Results	3
	1.2.1 Advanced Guidance Laws for Surface- to-Air Missiles	3
	1.2.2 Target Trajectory Estimation	3
	1.2.3 Adaptive Autopilots	3
	1.3 Report Organization	4
2	MISSILE GUIDANCE PROBLEM DISCUSSION	5
	2.1 Missile Subsystems	6
	2.2 Past Approaches for Missile Guidance	7
	2.3 Optimal Control Solution	9
	2.3.1 Dynamics	10
	2.3.2 Optimal Control Solution with Known Target Maneuvers and Missile States	10
	2.3.3 Optimal Control with Noisy Measure- ments and Estimated Initial State	11
	2.3.4 Optimal Control with Target Evasive Maneuvers	12
	2.3.5 Numerical Requirements for Optimal Guidance Laws	13
	2.4 Comparison of Optimal and Pronav Guidance	15
3	ADVANCED GUIDANCE LAWS FOR SURFACE-TO-AIR MISSILES	19
	3.1 Missile Model	19
	3.2 Optimal Solution	22
	3.3 Time-Scale Separation	24
	3.4 Simplified Optimal Control Solution Based on Separation of Time Scales	27
	3.4.1 Slowest Time Scale	27
	3.4.2 Altitude Dynamics	29
	3.4.3 Flight Path Angle Dynamics	30
	3.5 Alternate Energy State Formulation	30

TABLE of CONTENTS (CONTINUED)

Section

	3.5.1 Slowest Time Scale	32
	3.5.2 Altitude Dynamics	34
	3.5.3 Attitude Dynamics	38
4	TARGET DYNAMICS ESTIMATION AND ADAPTIVE TRAJECTORY TRACKING	41
	4.1 Equations of Motion	41
	4.2 Time-Scale Separation for State Estimation	42
	4.3 Recursive Maximum Likelihood (RML) Arma Identification	44
	4.4 Adaptive Lattice Algorithm for Trajectory Tracking	46
5	SUMMARY	51
	5.1 Advanced Guidance Laws	51
	5.2 Adaptive Target State Estimation and Tracking	51
	5.3 Future Research	52
	References	53
	APPENDIX A-PROPORTIONAL NAVIGATION GUIDANCE	55
	APPENDIX B-A QUASI-NEWTON ALGORITHM FOR ESTIMATION OF ARMA MODEL PARAMETERS	59
	APPENDIX C-THE ADAPTIVE LEAST-SQUARES PROBLEM	65
	APPENDIX D-ADAPTIVE CONTROL AND PREDICTION USING LATTICE STRUCTURES	69

SECTION 1

INTRODUCTION

Surface-to-surface, surface-to-air and air-to-surface tactical missiles of interest to the U. S. Army are being required to engage in increasingly demanding scenarios. The engagement requirements are expected to get even more exacting during the next decade [1]. These requirements can only be met with major improvements in guidance laws. The most difficult part of tactical missile guidance is the terminal-homing phase. Terminal homing generates missile commands to direct the warhead on a desired target or a specific region of the target.

Proportional navigation (pronav) has been used extensively for terminal guidance because of its success in conventional ground-to-ground and ground-to-air engagements. In addition, pronav is implemented by directly commanding missile acceleration components proportional to outputs of a gimballed seeker.

Several studies have shown that pronav is incapable of meeting the guidance requirements in the late 1980's and beyond. The three main reasons why pronav will not be acceptable for future missiles are: (1) improved accuracy requirement in conventional scenarios, (2) more demanding future engagements (e.g., guidance of anti-tactical ballistic missiles) and (3) increased stress on inexpensive seekers, gyros and accelerometers (e.g., strapdown seekers). Therefore, there is a need to develop advanced estimation, control and signal-processing techniques for Army tactical missiles.

1.1 SUMMARY OF APPROACH

The missile guidance problem is solved by decomposing it into three separate and simpler problems (Figure 1-1):

1. Missile guidance with known missile and target dynamics,
2. Estimation of target dynamics, and
3. Adaptive autopilot design.

The missile guidance solution is obtained using the singular perturbation extension to the optimal control solution. Target trajectory and dynamics are estimated by fitting an Autoregressive Moving Average (ARMA) model to the available measurement and updating the model parameters recursively in time. Stabilization of the missile with respect to random disturbances and tracking of the nominal trajectory provided by the optimal control solution are achieved by an autopilot based on an adaptive lattice algorithm.

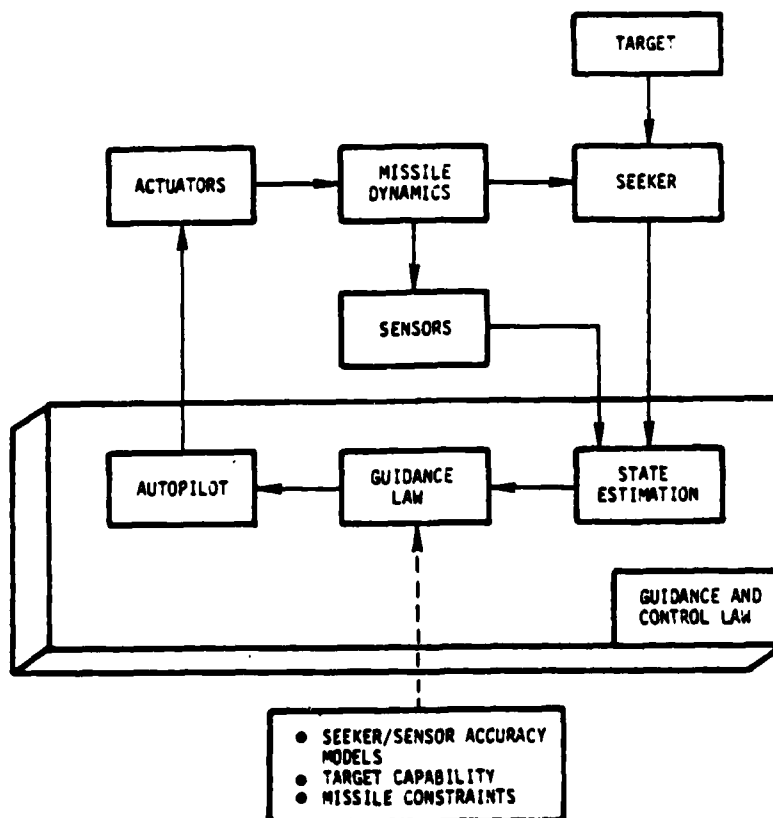


Figure 1-1 A Modern Control-Based Missile Guidance Law

1.2 RESULTS

1.2.1 Advanced Guidance Laws for Surface-to-Air Missiles

A singular perturbation guidance law has been developed for medium-range surface-to-air missiles [2]. This guidance law is a significant extension of a previously developed guidance law for short-range missiles [3]; in medium-range intercepts, the problem of energy management should be addressed in addition to homing guidance.

The mathematical formulation has been simplified by introducing separation of time scales. While time constants for medium-range intercepts are significantly different from those of short-range intercepts, the principle of time-scale separation is still applicable. The resulting simplified optimal control formulation requires solution to a set of nonlinear algebraic equations and to an initial value problem, all well-suited for on-board real-time implementation.

1.2.2 Target Trajectory Estimation

A recursive algorithm for estimation of ARMA model parameters from noisy samples has been developed. Application of this algorithm to parameter estimation problems has exhibited its fast convergence and unbiasedness in the presence of noise [4], even with short data records. The algorithm has two versions, a Recursive Maximum Likelihood (RML) form and a Recursive Prediction Error (RPE) form, both of which possess a parallel structure that makes them highly suitable for parallel-processing implementation.

1.2.3 Adaptive Autopilots

Lattice-form algorithms have been developed for fast, recursive identification and control of time-varying systems [5-7]. These algorithms have excellent numerical properties and a modular structure that makes them suitable for on-board real-time

implementation. Since the model parameters and the control input are updated at each measurement point, recursive lattice algorithms respond to changes in system dynamics faster than conventional, nonrecursive algorithms. This advantage makes recursive lattice algorithms a natural choice for incorporation in adaptive autopilots. They enable stabilization of the missile with respect to random disturbances with short time constants and tracking the nominal trajectory indicated by the optimal control solution.

1.3 REPORT ORGANIZATION

This report is organized as follows: Section 2 summarizes the missile guidance problem, Section 3 develops advanced guidance laws, and Section 4 describes the algorithms for target dynamics estimation and adaptive autopilot. Conclusions are given in Section 5.

SECTION 2

MISSILE GUIDANCE PROBLEM DISCUSSION

The missile-target dynamics are highly nonlinear partly because the equations of motion are best described in an inertial system, while aerodynamic/control forces and moments are represented in missile and target body axis systems. The linearization of the nonlinear equations of motion is complicated by fast changes in relative target and missile velocity vectors. Therefore, simplified estimation and control procedures for linear systems cannot be applied.

Proportional navigation (pronav) and other conventional guidance laws have been developed using classical control methods, based primarily on linear system formulations. It was also necessary to divide the overall problem into guidance and autopilot design to reduce the order of the problem. Since the missile guidance problem is highly nonlinear, time varying and of high order, classical methods are difficult to extend for improved guidance laws.

The modern control theory formulation can treat nonlinear systems with multiple inputs and multiple outputs. Modern control can also handle trajectory constraints (e.g., stability and energy management) and terminal requirements of small miss distance.

In general, tactical missile guidance has three phases, boost, midcourse and terminal. The boost and midcourse phase involve energy management, threat avoidance and navigation. The terminal phase is described by the seeker locked onto the target (or a region on the target). Guidance commands in the

terminal phase are generated to intercept the target. The terminal phase is the most demanding part of missile guidance. Our research is focused on improving missile performance during this critical phase.

2.1 MISSILE SUBSYSTEMS

Figure 2-1 shows important missile subsystems which impact guidance and control laws. Missile dynamics are driven by aerodynamic and propulsion system characteristics, control inputs and the ambient conditions. Guidance commands must be derived from sensor and seeker outputs. Sensors measure missile inertial states, usually angular rates and translation accelerations. The seeker measures components of target position and velocity relative to missile fixed-axis system.

The seeker derives information about the target using infra-red, laser, radar or other techniques and is often a major part of the missile cost. The seeker is subject to jamming and decoys. Good signal-processing techniques are often needed to minimize the degradation of target information due to countermeasures.

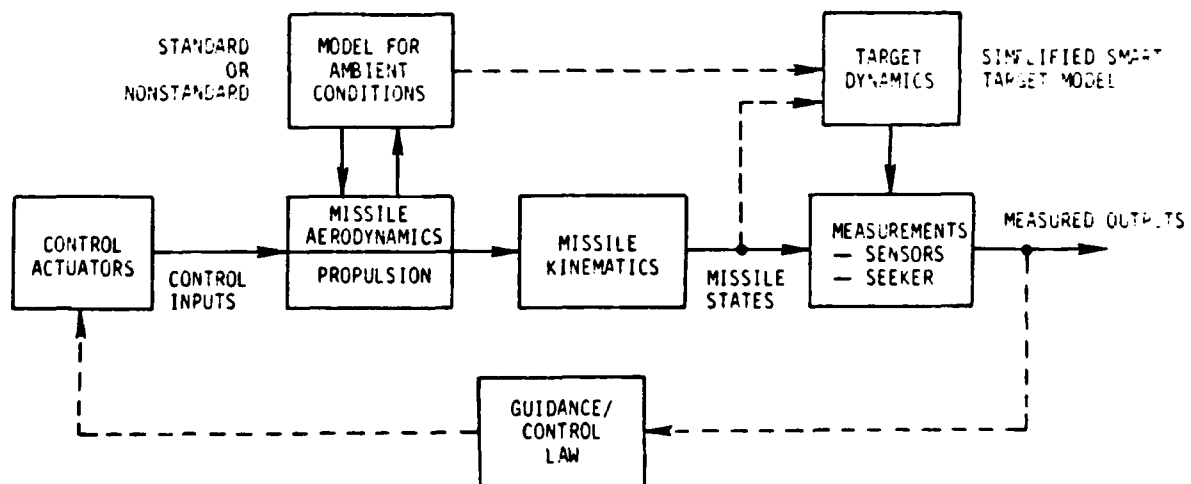


Figure 2-1 Major Missile Components

From the guidance viewpoint, seekers are either passive or active. Passive seekers measure the two components of line-of-sight rates or angular locations of the target in the missile axis system. Active seekers, in addition, provide measurements of range or range rate. Passive seekers are highly desirable because of lower cost and minimum vulnerability to jamming.

Each component in the overall model exhibits strong nonlinear behavior. Missile aerodynamics and propulsion models are nonlinear, particularly because of fast changes in speed, angle-of-attack and altitude. Sensors also have large nonlinear errors like scale factors and bias. Seeker outputs follow various trigonometric relationships causing additional nonlinear behavior. Table 2-1 shows typical actuator errors. These errors produce additional nonlinearities in missile dynamics.

2.2 PAST APPROACHES FOR MISSILE GUIDANCE

Many guidance laws have been used for the terminal phase in the past. Pursuit and proportional navigation (pronav) are the two most common techniques. More recently, pronav has been used almost exclusively in advanced missiles. Table 2-2 compares five classical approaches to missile guidance. Pronav will be our baseline technique because it has been most successful in previous work.

In pronav, the commanded missile acceleration is:

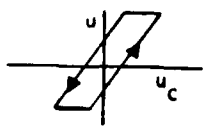
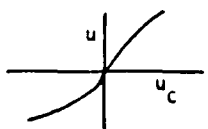
$$a_{mc} = \Lambda_n \frac{r\dot{\sigma}}{t_{go}}, \quad (2.1)$$

where r is the range, $\dot{\sigma}$ is the line-of-sight rate, t_{go} is the time to go and Λ_n is the navigation gain. In conventional pronav applications, t_{go} is approximated by

$$t_{go} = -r/\dot{r}, \quad (2.2)$$

such that

Table 2-1. Actuator Errors

ERROR	MODEL	EFFECT
Delay	$u(s) = \frac{u_c(s)}{1 + \tau s}$ or $u = u_c e^{-\tau s}$	<ul style="list-style-type: none"> - Reduces controller bandwidth - May cause instability in high gain controllers
Bias	$u = u_c - b$	<ul style="list-style-type: none"> - Produces steady state errors in commanded acceleration
Hysteresis (Backlash)		<ul style="list-style-type: none"> - Reduces controller bandwidth - May produce limit cycling
Bending or Flutter	 and $u = u_c + \text{random deflection}$	<ul style="list-style-type: none"> - Reduces control effectiveness at large commanded acceleration

u = control deflection; u_c = commanded control deflection

Table 2-2. Comparison of Classical Guidance Laws

GUIDANCE LAW	ADVANTAGES	DISADVANTAGES
Command-to-Line-of-Sight Guidance	No terminal seeker required	<ul style="list-style-type: none"> • Very inaccurate against moving targets and with winds • Data link required
Pursuit	<ul style="list-style-type: none"> • Noise insensitive • Easy to use with strapdown seekers 	<ul style="list-style-type: none"> • Inaccurate against moving targets and with winds
Proportional	Accurate against constant velocity targets	<ul style="list-style-type: none"> • Inaccurate against accelerating targets • Stability is sensitive to noise
Pursuit + Pronav	Between 2 and 3 in terms of accuracy	<ul style="list-style-type: none"> • Between 2 and 3
Dynamic Lead	<ul style="list-style-type: none"> • Between 2 and 3 in terms of accuracy • Easy to use with strapdown seekers 	<ul style="list-style-type: none"> • Between 2 and 3 • Stability problems if transition to pronav occurs when significant noise is present

$$a_{mc} = -\Lambda_n \dot{r} \dot{\sigma} .$$

Navigation gain of between 2 and 4 is considered desirable. To achieve this navigation gain, both $\dot{\sigma}$ and \dot{r} should be used. $\dot{\sigma}$ is available from passive as well as active seekers. \dot{r} , however, is measured only by active seekers. Conventional implementations of pronav with constant navigation ratio, therefore, require active seekers. The variation in navigation ratio for stationary or constant speed targets is, nevertheless, small even with passive seekers.

Characteristics of pronav guidance laws are discussed in Appendix A. Appendix A indicates that errors occur in pronav guidance because of the following:

1. Instability of pronav guidance law prior to impact,
2. Target maneuvers and variations in target speed,
3. Variations in missile speed,
4. Missile dynamics and combined missile/autopilot dynamics, and
5. Sensor errors and seeker saturation.

Modern control and estimation methods based on advanced guidance algorithms can overcome the problems indicated above. The next section discusses the general modern control theory formulation.

2.3 OPTIMAL CONTROL SOLUTION

Three cases for optimal control are considered: (1) known target maneuvers and missile states, (2) known-target maneuvers but estimated missile states and (3) evasive intelligent targets. The resulting numerical problems are shown for each missile guidance solution.

2.3.1 Dynamics

The dynamics of a missile-target engagement may be described by dynamic equations of the form:

$$\begin{aligned}\dot{x} &= f(x, u, a_T) , & x(0) &= x_0 , \\ 0 &\leq t \leq t_f , & & (2.3)\end{aligned}$$

where x is the state vector consisting of: (1) components of relative position, (2) components of relative velocity, (3) missile orientation angles, (4) missile angular rates, and (5) actuator and sensor states. u is the vector of control surface deflections and a_T is the target acceleration vector.

2.3.2 Optimal Control Solution With Known Target Maneuvers and Missile States

The optimal control solution to missile guidance is based on minimizing the following function of states and inputs

$$J = S(x(t_f)) + \int_0^{t_f} \ell(x, u, a_T) dt . \quad (2.4)$$

The first term in Eq. 2.4 defines terminal requirements for target intercept. This term decreases as the relative terminal distance between the missile and target decreases. Requirements for relative terminal angular orientation for improved charge detonation may also be included in $S(x(t_f))$. The second term specifies a preferred missile trajectory. This term may be used to manage energy, to satisfy seeker constraints or simply to minimize flight time. Minimization of flight time, for example, is achieved by setting ℓ to unity and $S(x(t_f)) = 0$. The final time, t_f , may be constrained but is usually free in missile guidance problems.

The optimization problem is solved by using a Lagrange variable λ and a Hamiltonian H . The equations leading to the optimal solution are:

$$H = L(x, u, a_T) + \lambda^T f(x, u, a_T) , \quad (2.5)$$

$$\dot{\lambda} = - \left(\frac{\partial H}{\partial x} \right)^T = - \left(\frac{\partial f}{\partial x} \right)^T \lambda - \left(\frac{\partial L}{\partial x} \right)^T , \quad \lambda(t_f) = \frac{\partial S}{\partial x} , \quad (2.6)$$

and

$$\left(\frac{\partial H}{\partial u} \right)^T = \left(\frac{\partial L}{\partial u} \right)^T + \left(\frac{\partial f}{\partial u} \right)^T \lambda = 0 , \quad (2.7)$$

$$H(t_f) = 0 . \quad (2.8)$$

Eqs. 2.3 and 2.6 are coupled. The control is obtained from Eq. 2.7 and the final time is obtained from Eq. 2.8. The initial condition is defined for states x and the final condition for variables λ . Therefore, the computation of u requires solution to a two-point boundary value problem (TPBVP) represented by Eqs. 2.3 - 2.8. The optimal control can be determined if x_0 and $a_T(t)$ are known and the cost functional is defined.

Therefore, with known initial conditions and target maneuvers, modern control requires a TPBVP to be solved for missile guidance commands. The missile guidance problem becomes more complex when the initial condition must be estimated from seeker outputs and the target may perform evasive maneuvers.

2.3.3 Optimal Control with Noisy Measurements and Estimated Initial State

Table 2-3 shows outputs of passive and active seekers. Clearly not all states are measured. The measurements are also noisy.

A state estimator (e.g., Kalman filter) is required to determine the current state vector needed for optimal missile guidance. The extended Kalman filter formulation may be used. For n states, n differential equations are required for the

Table 2-3. Output of Missile Seekers

KIND	OUTPUT	RELATIONSHIP TO MISSILE STATES
PASSIVE (Infrared, Optical, Passive Radar)	Two Line-of-Sight Angles	Ratio of Lateral Target Position Components to Range (in Missile Axis Systems)
	Two Line-of-Sight Rates	Time Derivatives of Above
ACTIVE (Active Radar)	Range	Sum of Square of Relative Position Components
	Range Rate	Time Derivative of Range

state estimate. In addition, $n(n+1)/2$ equations must be propagated for the covariance matrix and the Kalman gain. Since the number of equations for the covariance matrix is much larger than the number of equations for the state estimate, there is a significant payoff in simplifying the covariance equations.

In linear systems, the error covariance matrix is independent of the measurements. Therefore, the estimation error does not depend upon the applied input. However, in nonlinear systems the measurements affect estimation accuracy. The error in state estimates may, therefore, be reduced by appropriate application of inputs. This leads to a dual control formulation, where the inputs provide guidance as well as improvement in estimation accuracy. The exact solutions to these problems are quite complex and simplifications are necessary for real-time implementations.

2.3.4 Optimal Control With Target Evasive Maneuvers

In the previous two sections, we assumed that future target acceleration time trace, a_T , is known. If the target is capable of performing evasive maneuvers, the future target acceleration depends upon the missile flight path. At any time point, the target will perform the most desirable maneuver to void the missile.

The assumption of target evasive maneuvers leads to a differential game formulation. The target will determine its

commanded acceleration by maximizing the same or similar penalty functional that the missile wants to minimize. The optimization problem may be stated as

$$\max_{a_T} \min_u J . \quad (2.9)$$

This optimization problem requires the solution to Eqs. 2.9 - 2.8 and the following equation:

$$\frac{\partial H}{\partial a_T} = 0 . \quad (2.10)$$

Note that the resulting TPBVP is even more difficult to solve than the one with known target maneuvers due to the additional constraint given by Eq. 2.10.

2.3.5 Numerical Requirements for Optimal Guidance Laws

Figure 2-2 is a flowchart for a missile guidance law based on modern control theory. Note that a state estimation step is required. In addition, the guidance law needs more information than with pronav.

Table 2-4 shows the various optimal control formulations for missile guidance and the resulting numerical procedures needed for their solution. The numerical algorithms are difficult to implement. Therefore, simplifications which do not compromise accuracy are needed for effective missile guidance laws.

Table 2-5 indicates possible approaches for simplifying the numerical problems associated with the optimal control solution. Because the basic missile-target dynamics are nonlinear, each of these simplifications must be developed specifically for the missile guidance law. A singular perturbation simplification technique has already been developed. This

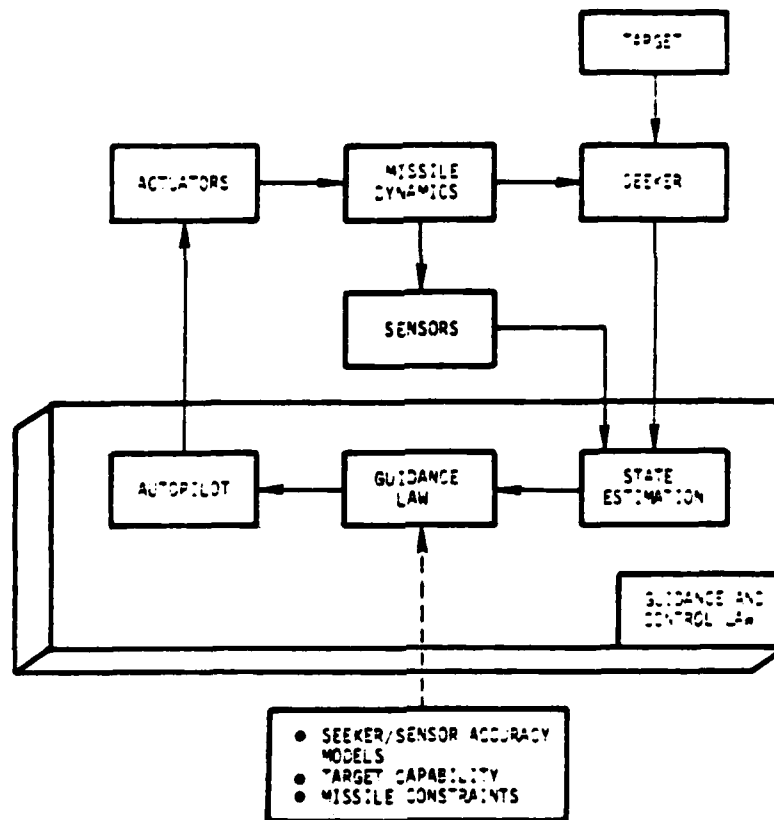


Figure 2-2. A Modern Control-Based Missile Guidance Law

Table 2-4. Optimal Control Formulations for Missile Guidance

FORMULATION	OPTIMAL CONTROL PROBLEM	NUMERICAL PROCEDURES
Known Target Maneuvers and Missile States	Optimal Control Law with Terminal Constraints and Free Terminal Time	Two-Point Boundary Value Problem with Path Constraints
Noisy Seeker Measurements	Kalman Filter	Propagation of Differential Equations; Computation of Derivatives
	Dual Control	Stochastic Two-Point Boundary Value Problem
Target Evasive Maneuvers	Differential Game (Known State)	Difficult Boundary Value Problem
	Dual Differential Game (Unknown State)	Difficult Stochastic Boundary Value Problem and Kalman Filter

Table 2-5. Simplification of Optimal Control Numerical Procedures

NUMERICAL PROCEDURES	SIMPLIFICATION/APPROXIMATION	RESULTING NUMERICAL PROBLEM
Two-Point Boundary Value Problem with Path Constraints (Optimal Control)	<ul style="list-style-type: none"> • Linearization • Separation of time scales (singular perturbation) 	<ul style="list-style-type: none"> • Initial value problem • Algebraic problem • Partially coupled solution
Propagation of Differential Equations; Computation of Derivatives (Kalman Filter)	<ul style="list-style-type: none"> • Simplify Kalman gain or compute it off-line • Time scale separation 	<ul style="list-style-type: none"> • Reduced number of differential equations • Partially coupled solution
Stochastic Two-Point Boundary Value Problem (Dual Control)	<ul style="list-style-type: none"> • Define a class of test inputs 	<ul style="list-style-type: none"> • Convert to an optimal control problem
Difficult Boundary Value Problem (Differential Game)	<ul style="list-style-type: none"> • Parameterize guidance law • Reachable sets • Command constraint guidance 	<ul style="list-style-type: none"> • Direct solution for missile guidance and target evasion • Simplified optimal control solution
Difficult Stochastic Boundary Value Problem and Kalman Filter (Dual Differential Game)	<ul style="list-style-type: none"> • Define a class of test inputs 	<ul style="list-style-type: none"> • Convert to an optimal control solution

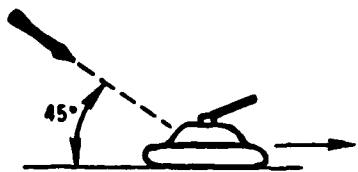
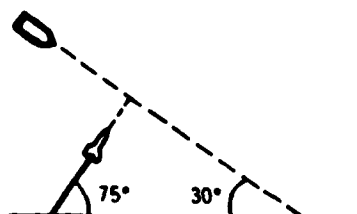
technique is described in [3]. The singular perturbation guidance algorithm for anti-tactical ballistic missiles (ATBMs) is shown in Section 3. The next section (Section 2.4) contains a comparison of optimal and pronav guidance in one particular ATBM scenario.

2.4 COMPARISON OF OPTIMAL AND PRONAV GUIDANCE

We compare miss distances for an anti-tank missile and for an ATBM due to fore-aft target acceleration. The two scenarios are summarized in Table 2-6. The ATBM has higher speed and faster dynamic response. The pronav gain in each case is 3.

Miss in pronav occurs because of the inability of the missile to track slowing ballistic missiles and in the latter part of the flight due to missile instability. The target tank, in the first scenario, increases miss distance by stepping on the brakes.

Table 2-6 Description of the Engagement Scenario

	SCENARIO I ANTI-TANK MISSILE	SCENARIO II ANTI-TACTICAL BALLISTIC MISSILE
		
Missile Speed (ft s ⁻¹)	500	4,000
Target Speed (ft s ⁻¹)	40	10,000
φ_t	45°	20°
φ_m	3°	305°
z_m	0.5	0.5
ω_m (s ⁻¹)	3	20
Λ_n	3	3

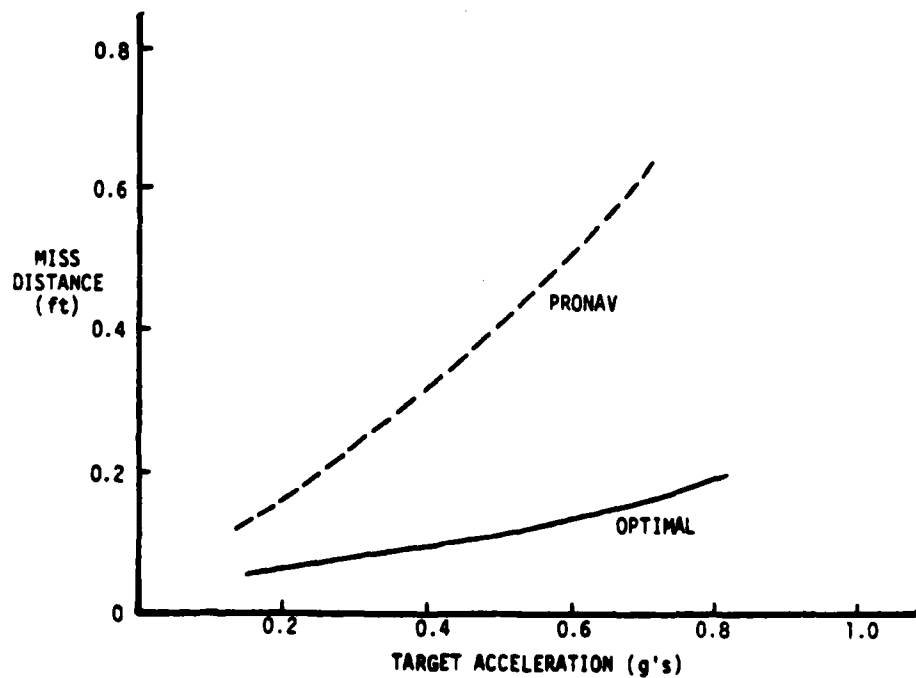


Figure 2-3 Comparison of Pronav and Optimal Guidance Laws for an Anti-Tank Missile

Figure 2-3 shows miss distance as a function of tank deceleration (the maximum deceleration is expected to be 0.5 g). The effect of instabilities is significantly amplified in the presence of noise. Therefore, the control input is often set to zero in the unstable region. If the control input is set to zero half-way through the instability, the miss distances of Figure 2-3 will approximately double.

Similar plots for the ATBM engagement are shown in Figure 2-4. Most of this miss distance results because pronav does not use the entire missile capabilities. An optimal control law gives miss distance as shown by the solid line. The optimal control law is able to achieve this improvement because it starts applying acceleration early in the trajectory (Figure 2.5).

The results presented in this section show that in one ATBM engagement scenario, the pronav guidance cannot meet performance requirements, while singular perturbation does adequately well.

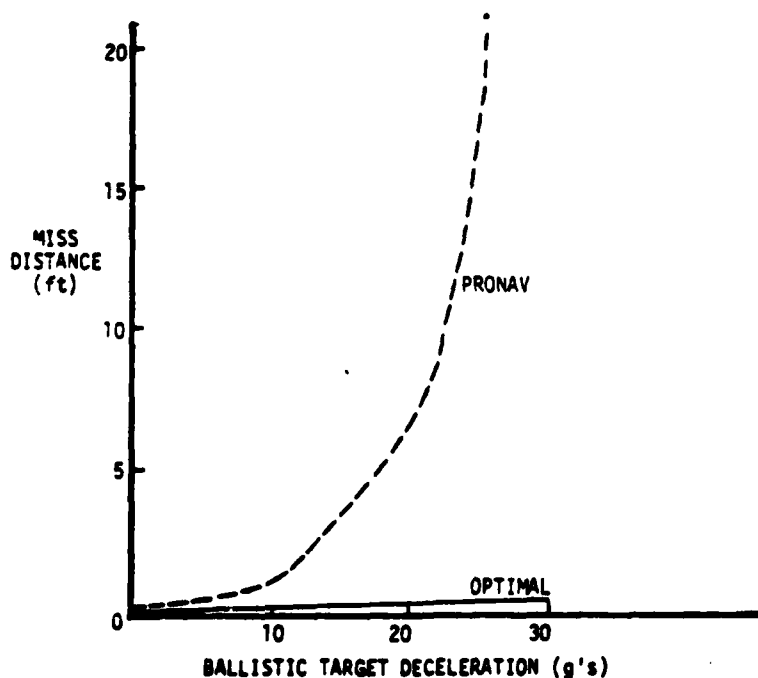


Figure 2-4 Comparison of Pronav and Optimal Guidance Laws for ATBM

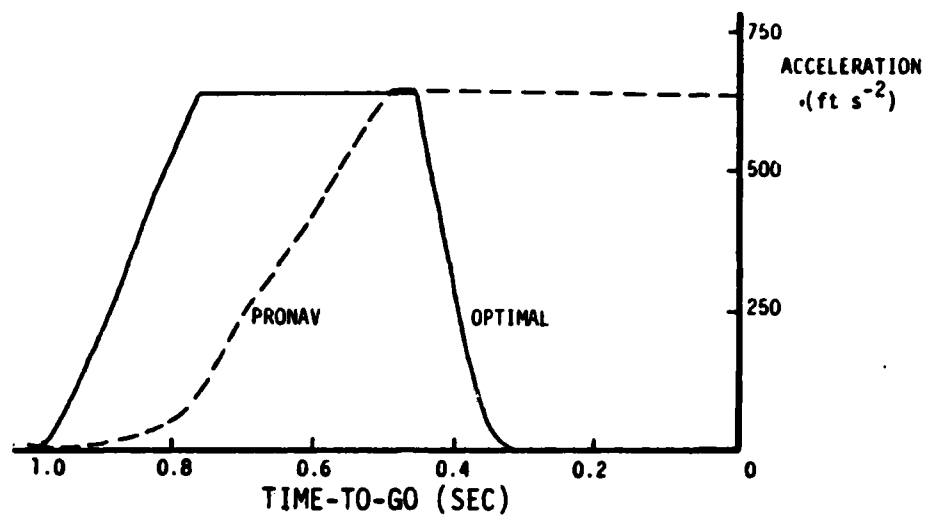


Figure 2-5 Comparison of Missile Acceleration Profiles for Pronav Guidance and Optimal Guidance

SECTION 3

ADVANCED GUIDANCE LAWS FOR SURFACE-TO-AIR MISSILES

This section describes a model for an advanced aircraft missile and develops an optimal control formulation for improved guidance laws. A singular perturbation extension to the optimal control solution is presented. The time constants selected in the singular perturbation method are based on physical variables as shown later. The resulting solution involves a set of non-linear algebraic equations. Two formulations have been used -- one based on the kinematic state and the other using the energy state.

3.1 MISSILE MODEL

With \vec{x}_m , \vec{v}_m and \vec{a}_m , missile position, velocity and aerodynamic acceleration vector, respectively, the missile dynamic model in the Cartesian coordinate system is

$$\dot{\vec{x}}_m = \vec{v}_m, \quad (3.1)$$

$$\dot{\vec{v}}_m = \vec{a}_m - \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} g. \quad (3.2)$$

$\vec{x}_m = (x, y, h)^T$, $\vec{v}_m = (u, v, w)^T$ and $\vec{a}_m = [a_x, a_y, a_z]^T$. The acceleration component along the velocity vector is defined, because missiles in Army inventory do not have thrust control (the formulation can be modified, if necessary).

$$\frac{1}{V} \vec{v}_m \cdot \vec{a}_m = \frac{(T-D)}{m}, \quad (3.3)$$

or

$$u a_x + v a_y + w a_z = \frac{(T-D)V}{m} \quad (3.4)$$

m is a function of time and its time history is known. Thus, we can consider a_x , a_y and a_z as our control variables with the constraint of (3.4).

To simplify the derivation of the guidance law, we convert the velocity equations into a total speed and two flight angle equations. Equations 3.1 and 3.2, thus, become

$$\dot{x} = V \cos \gamma \cos \phi, \quad (3.5)$$

$$\dot{y} = V \cos \gamma \sin \phi, \quad (3.6)$$

$$\dot{h} = V \sin \gamma, \quad (3.7)$$

$$\dot{V} = (T-D)/m - g \sin \gamma, \quad (3.8)$$

$$\dot{\phi} = L \sin \sigma / m V \cos \gamma, \quad (3.9)$$

where

$$\dot{\gamma} = (L \cos \sigma - mg \cos \gamma) / m V. \quad (3.10)$$

ϕ and γ are flight path angles in the horizontal and the vertical planes (see Figure 3-1). $L \sin \sigma$ and $L \cos \sigma$ are lift components in the x - y and vertical planes, respectively. Thus, the two control variables are given explicitly in this formulation.

The thrust time history is predefined. The drag is written as follows

$$D = \frac{1}{2} \rho(h) V^2 S C_A, \quad (3.11)$$

$$C_A = C_{A_0} + C_{A_\alpha} |\alpha| + C_{A_{\alpha^2}} \alpha^2, \quad (3.12)$$

$$q = \frac{1}{2} \rho(h) V^2. \quad (3.13)$$

α is the total angle-of-attack and the density ρ is a function

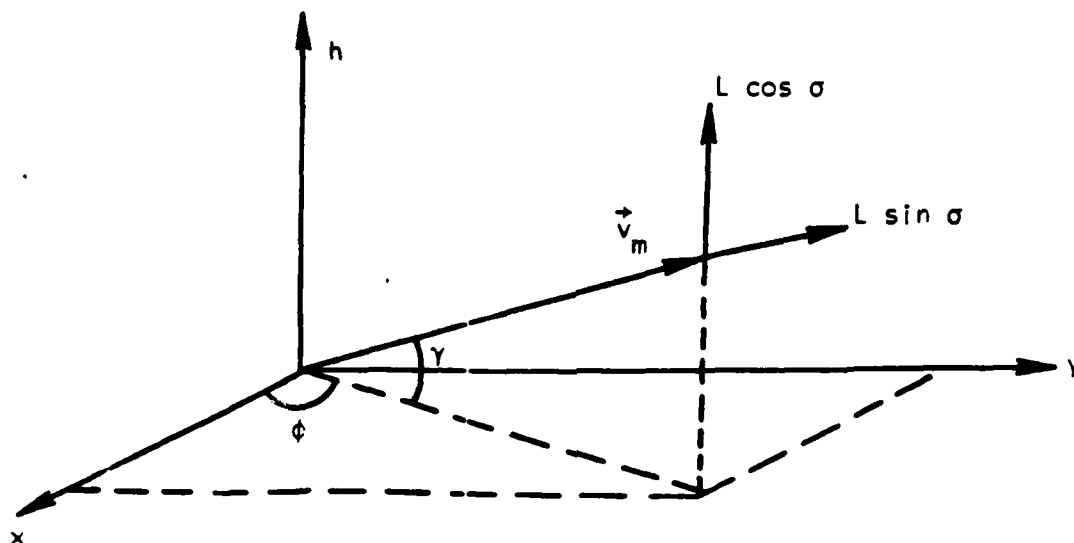


Figure 3-1 Axis System for Missile Dynamics

of altitude. The total angle-of-attack is related to total lift as follows

$$L = \frac{1}{2} \rho(h) v^2 C_N \quad (3.14)$$

The target position velocity and acceleration vectors are related to each other

$$\dot{\vec{x}}_T = \vec{v}_T \quad (3.15)$$

$$\dot{\vec{v}}_T = \vec{a}_T \quad (3.16)$$

The final condition for missile intercept is

$$\vec{x}_m(t_f) = \vec{x}_T(t_f) \quad (3.17)$$

where t_f is free. The optimal control solution will be based on the minimization of

$$J = \int_0^{t_f} \mathcal{L}(\vec{x}_m, \vec{v}_m, \vec{a}_m) dt \quad (3.18)$$

Typically \mathcal{L} is a function of drag, total speed, energy loss rate and total flight time. If the optimization criterion were minimum time, the performance index would be

$$\mathcal{L} = 1 \quad (3.19)$$

and for the minimum energy loss solution

$$\mathcal{L} = D \cdot V \quad (3.20)$$

In our discussion, it is assumed that $\vec{x}_T(t)$ is known, based on an estimate of the target motions.

3.2 OPTIMAL SOLUTION

The solution to the optimal control problem is obtained by defining a Hamiltonian as follows. The Lagrange multiplier method for the optimal solution gives the following Hamiltonian for this problem

$$\begin{aligned} H = & \lambda_x V \cos \gamma \cos \phi + \lambda_y V \cos \gamma \sin \phi + \lambda_h V \sin \gamma \\ & + \lambda_v [(T-D)/m - g \sin \gamma] + \lambda_\phi L \sin \sigma / (mV \cos \gamma) \\ & + \lambda_\gamma (L \cos \sigma - mg \cos \gamma) / (mV) + \mathcal{L} . \end{aligned} \quad (3.21)$$

$\lambda_x, \lambda_y, \lambda_h, \lambda_v, \lambda_\phi, \lambda_\gamma$ are the various Lagrange parameters and \mathcal{L} is a general cost functional. Since the Hamiltonian does not depend on x or y

$$\dot{\lambda}_x = \dot{\lambda}_y = 0 . \quad (3.22)$$

The other four Lagrange variables are

$$\dot{\lambda}_h = -\partial H / \partial h , \quad \lambda_h(t_f) \text{ free} , \quad (3.23)$$

$$\dot{\lambda}_V = -\partial H / \partial V , \quad \lambda_V(t_f) = 0 , \quad (3.24)$$

$$\dot{\lambda}_\phi = -\partial H / \partial \phi , \quad \lambda_\phi(t_f) = 0 , \quad (3.25)$$

$$\dot{\lambda}_\gamma = -\partial H / \partial \gamma , \quad \lambda_\gamma(t_f) = 0 . \quad (3.26)$$

The optimality conditions give

$$\begin{aligned} \frac{\partial H}{\partial L} = & -\frac{\lambda_V}{m} \frac{\partial D}{\partial L} + \lambda_\phi \sin \sigma / (mV \cos \gamma) \\ & + \lambda_\gamma \cos \sigma / (mV) + \frac{\partial \mathcal{L}}{\partial L} = 0 , \end{aligned} \quad (3.27)$$

$$\frac{\partial H}{\partial \sigma} = \lambda_\phi L \cos \sigma / (mV \cos \gamma) - \lambda_\gamma L \sin \sigma / (mV) = 0 . \quad (3.28)$$

Note that neither D nor \mathcal{L} should depend on σ . Thus,

$$\tan \sigma = \frac{\lambda_\phi}{\lambda_\gamma \cos \gamma} . \quad (3.28)$$

Since the final time is free,

$$H(t_f) = 0 . \quad (3.30)$$

Note that H is an explicit function of time through T and m . Thus, H is not zero throughout.

The unknowns are λ_x , λ_y , L and σ and the time histories of λ_h , λ_V , λ_ϕ , and λ_γ . Thus, six forward and four backward differential equations need to be solved in a two-point boundary value problem (TPBVP).

3.3 TIME-SCALE SEPARATION

The six missile dynamics equations are scaled as follows

$$\frac{r}{V} \frac{d(x/r)}{dt} = \cos \gamma \cos \phi , \quad (3.31)$$

$$\frac{r}{V} \frac{d(y/r)}{dt} = \cos \gamma \sin \phi , \quad (3.32)$$

$$\frac{h_{\max}}{V} \frac{d(h/h_{\max})}{dt} = \sin \gamma . \quad (3.33)$$

The time constant associated with the total speed equation is determined by substituting for α in terms of lift in the drag equations (Eqs. 3.11 - 3.14)

$$\frac{dV}{dt} = \frac{T}{m} - \frac{1}{m} \left[QsC_{A_0} + \frac{C_{A_\alpha}}{C_{N_\alpha}} L + \frac{C_{A_\alpha}^2 L^2}{QsC_{N_\alpha}^2} \right] - \frac{wg}{V} . \quad (3.34)$$

The total speed equation has two parts. The first term is known and can be large when the thrust is on. The second term depends on flight condition and control input and, thus, controls the dynamics of the total speed equation. The drag term varies from fractions of a g to a few g's in most medium-range missiles. The gravity term is always less than one g (because $w/V < 1$). Thus, dividing by g the right-hand side becomes of unit order. The time constant associated with this equation could change if the missile drag was larger by a factor at high angle-of-attack.

$$\frac{V}{g} \left(\frac{1}{V} \frac{dV}{dt} \right) = \frac{(T-D)}{mg} - \frac{w}{V} . \quad (3.35)$$

Similarly, a time constant can be developed for the total energy equation

$$E = \frac{V^2}{2g} + h ,$$

$$\frac{dE}{dt} = \frac{(T-D)}{mg} V . \quad (3.36)$$

Since D is of order of mg , the nondimensional equation for E is

$$\frac{V}{g} \frac{1}{E} \frac{dE}{dt} = \frac{2}{1 + \frac{2gh}{V^2}} \frac{(T-D)}{mg} . \quad (3.37)$$

Thus, the total energy equation has essentially the same time constant as the speed equation.

The time constants associated with ϕ and γ are obtained as follows

$$\frac{mV}{L_{\max}} \frac{d\phi}{dt} = \frac{L}{L_{\max}} \frac{\sin \sigma}{\cos \gamma} , \quad (3.38)$$

$$\frac{mV}{L_{\max}} \frac{d\gamma}{dt} = \frac{L}{L_{\max}} \cos \alpha + \frac{mg \cos \gamma}{L_{\max}} . \quad (3.39)$$

Thus, the time constant associated with flight path angles is

$$V/a_{\max} . \quad (3.40)$$

The missile lateral acceleration commands are generated by the autopilot. The particular separation of time scales for short-range missiles was discussed previously. Table 3-1 shows that the time scales for missiles with significant cruise phase (medium-range intercepts) could be significantly different. A clear separation of time scales associated with various states is clear.

<u>Slowest:</u>	position components, x and y , and total speed equation,
<u>Slow:</u>	altitude, h ,
<u>Fast:</u>	flight path angles, and
<u>Very fast:</u>	autopilot states.

Table 3-1 Time Constants Associated With Various States During Midcourse

	COMPONENTS	TIME CONSTANT	VALUE (s)
Position	x	r/V	40
	y	r/V	40
	h	h_{\max}/V	6
Velocity	Total speed (or energy)	V/g	50
	Flight path angles	V/a_{\max}	1.67
Acceleration	L Orientation	Autopilot time constants	0.6 to 0.05

$r = 20,000 \text{ m}$, $h_{\max} = 3,000 \text{ m}$, $V = 500 \text{ msec}^{-1}$, $a_{\max} = 30 \text{ g's}$

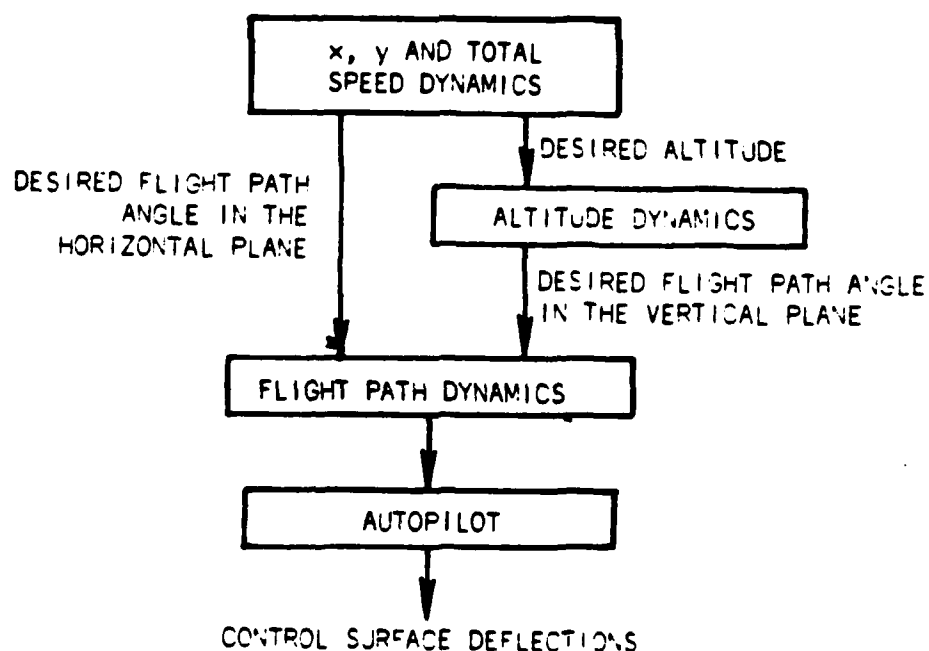


Figure 3-2 Schematic of Singular Perturbation Guidance Logic for Midcourse Phase

Note that the distribution of time scales is different from that realized for the end-game problem.

The approach to obtain guidance commands using singular perturbation methodology during midcourse phase is shown in Figure 3-2.

3.4 SIMPLIFIED OPTIMAL CONTROL SOLUTION BASED ON SEPARATION OF TIME SCALES

We will solve the guidance problem in four parts, each part corresponding to one of the time scales.

3.4.1 Slowest Time Scale

x , y and V are the slowest variables. Since h , ϕ and γ are faster, those equations may be considered to be in equilibrium. Therefore (the variables in the slowest time scales are denoted by superscript "1"),

$$\gamma^1 = \sigma^1 = 0, \quad (3.41)$$

$$L^1 = mg. \quad (3.42)$$

Thus, the Hamiltonian simplifies to

$$H^1 = \lambda_x^1 V^1 \cos \phi^1 + \lambda_y^1 V^1 \sin \phi^1 + \lambda_V^1 (T-D)^{1/2} / m + \mathcal{L}. \quad (3.43)$$

The adjoint equations for λ_h and λ_ϕ give

$$\begin{aligned} \frac{\partial H}{\partial h} = 0 = & -\frac{\lambda_V^1}{m} \frac{\partial D^1}{\partial h} + \frac{\partial \mathcal{L}}{\partial h} \\ & + (\lambda_x^1 \cos \phi^1 + \lambda_y^1 \sin \phi^1)(-g/V^1) \end{aligned} \quad (3.44)$$

$$\frac{\partial H}{\partial \phi} = 0 = -\lambda_x^1 V^1 \sin \phi^1 + \lambda_y^1 V^1 \cos \phi^1. \quad (3.45)$$

because

$$v^1 = v^2 - 2gh$$

Eq. 3.45 gives

$$\tan \phi^1 = \lambda_y^1 / \lambda_x^1 \quad (3.46)$$

Since λ_x^1 and λ_y^1 are known to be constants, the flight path in the horizontal plane for the slowest dynamics is a straight line. The straight line must joint the horizontal projections of the desired current and final conditions. In addition, since $H^1(t_f)$ and $\lambda_V^1(t_f)$ are zero,

$$\lambda_x^1 v^1(t_f) \cos \phi^1 + \lambda_y^1 v^1(t_f) \sin \phi^1 + \ell^1(t_f) = 0 \quad (3.47)$$

Eqs. 3.46 and 3.47 give

$$\lambda_x^1 = \frac{-\ell^1(t_f) \cos \phi^1}{v^1(t_f)} \quad (3.48)$$

$$\lambda_y^1 = \frac{-\ell(t_f) \sin \phi^1}{v^1(t_f)} \quad (3.49)$$

The adjoint equation for λ_V is

$$\begin{aligned} \dot{\lambda}_V^1 &= \lambda_x^1 \cos \phi^1 + \lambda_y^1 \sin \phi^1 - \frac{\lambda_V^1}{m} \frac{\partial D^1}{\partial V} + \frac{\partial \ell}{\partial V} \\ &= \frac{\partial}{\partial V} - \frac{\ell^1(t_f)}{v^1(t_f)} - \frac{\lambda_V^1}{m} \frac{\partial D^1}{\partial V}, \quad \lambda_V^1(t_f) = 0 \end{aligned} \quad (3.50)$$

Eqs. 3.44 and 3.50 give the optimal altitude in slowest time-scale approximation. This still requires the solution to a first-order TPBVP. This requirement could be removed if the Hamiltonian was not an explicit function of time.

3.4.2 Altitude Dynamics

To solve for the optimization problem in the altitude dynamics time scale, λ_x , λ_y and λ_v can be used from the previous solution. Since flight path angle dynamics are still faster than altitude dynamics, setting $\dot{\phi}$ and $\dot{\gamma}$ to zero we get (variables in altitude dynamics time scales are written with superscript "2")

$$\sigma^2 = 0, \quad L^2 = mg \cos \gamma^2. \quad (3.51)$$

The Hamiltonian can thus be modified to

$$\begin{aligned} H^2 = & \lambda_x^1 v^1 \cos \gamma^2 \cos \phi^1 + \lambda_y^1 v^1 \cos \gamma^2 \sin \phi^1 \\ & + \lambda_h^2 v^1 \sin \gamma^2 + \lambda_v^1 [(T-D^2)/m - g \sin \gamma^2] + \ell. \end{aligned} \quad (3.52)$$

The adjoint equation λ_γ gives

$$\begin{aligned} \frac{\partial H}{\partial \gamma} = 0 = & -\lambda_x^1 v^1 \sin \gamma^2 \cos \phi^1 - \lambda_y^1 v^1 \sin \gamma^2 \sin \phi^1 \\ & + (\lambda_h^2 v^1 - g \lambda_v^1) \cos \gamma^2 = 0. \end{aligned} \quad (3.53)$$

ℓ is not likely to be a function of γ . Thus,

$$\tan \gamma^2 = \frac{(\lambda_h^2 v^1 - g \lambda_v^1)}{\ell(t_f)}. \quad (3.54)$$

The adjoint equation for λ_h^2 is

$$\frac{d\lambda_h^2}{dt} = \frac{\lambda_v^1}{m} \frac{\partial D^2}{\partial h} - \frac{\partial \ell}{\partial h}, \quad \lambda_h^2(t_f) \text{ is free}. \quad (3.55)$$

$\lambda_h^2(t_f)$ may be determined from the relationship that $H^2(t_f)$ is zero. Thus, the optimal flight path angle may be determined from (3.54) and (3.55).

3.4.3 Flight Path Angle Dynamics

The approach to compute flight path angle dynamics has already been solved. Reference [1] provides an approach to direct the missile from an initial flight path to the desired flight path angles (ϕ, γ) . As has been shown previously, lift is applied perpendicular to the plane containing the initial and the desired velocity vector. The magnitude of the lift is proportional to the square root of the angle through which the flight path must be changed (see below).

3.5 ALTERNATE ENERGY STATE FORMULATION

The optimality equations and the corresponding simplifications have been obtained using the energy state to replace the total speed state in the missile equations of motion. This offers certain computational advantages as will be seen in the optimality equations. With E replacing V , the state equations are

$$\dot{x} = V \cos \gamma \cos \phi \quad (3.56)$$

$$\dot{y} = V \cos \gamma \sin \phi \quad (3.57)$$

$$\dot{h} = V \sin \gamma \quad (3.58)$$

$$\dot{E} = V(T-D)/mg \quad (3.59)$$

$$\dot{\phi} = (L \sin \sigma)/mV \cos \gamma \quad (3.60)$$

$$\dot{\gamma} = (L \cos \sigma - wg \cos \gamma)/mV \quad (3.61)$$

The performance index will be written as a combination of the flight time and the energy loss during flight.

$$J = \int_0^{t_f} (1-\xi) dt + \xi \frac{E(t_f)}{E_0} \quad (3.62)$$

E_0 is a normalizing factor and represents nominal energy loss per unit time. General performance indices can also be studied.

The Hamiltonian for this problem is

$$\begin{aligned}
 H = & \lambda_x V \cos \gamma \cos \phi + \lambda_y V \cos \gamma \sin \phi + \lambda_h V \sin \gamma \\
 & + \lambda_E V(T-D)/mg + (\lambda_\phi L \sin \sigma)/(mV \cos \gamma) \\
 & + \lambda_\gamma (L \cos \sigma - mg \cos \gamma)/mV + (1 - \xi)
 \end{aligned} \tag{3.63}$$

The optimality equations are

$$\dot{\lambda}_x = \dot{\lambda}_y = 0 \tag{3.64}$$

$$\dot{\lambda}_h = - \partial H / \partial h \quad \lambda_h(t_f) = \text{free} \tag{3.65}$$

$$\dot{\lambda}_E = - \partial H / \partial E \quad \lambda_E(t_f) = \xi / E_0 \tag{3.66}$$

$$\dot{\lambda}_\phi = - \partial H / \partial \phi \quad \lambda_\phi(t_f) = \text{free} \tag{3.67}$$

$$\dot{\lambda}_\gamma = - \partial H / \partial \gamma \quad \lambda_\gamma(t_f) = \text{free} \tag{3.68}$$

The optimality equations are

$$\frac{\partial H}{\partial L} = - \frac{\lambda_E V}{mg} \frac{\partial D}{\partial L} + \frac{\lambda_\phi \sin \sigma}{mV \cos \gamma} + \frac{\lambda_\gamma \cos \sigma}{mV} = 0 \tag{3.69}$$

$$\frac{\partial H}{\partial \sigma} = \frac{\lambda_\phi L \cos \sigma}{mV \cos \gamma} - \frac{\lambda_\gamma L \sin \sigma}{mV} = 0 \tag{3.70}$$

or

$$\tan \sigma = \frac{\lambda_\phi}{\lambda_\gamma \cos \gamma} \tag{3.71}$$

Since the final time is free

$$H(t_f) = 0 \tag{3.72}$$

Again a four time scale solution is sought. The ordering of the

time scales is as follows:

x, y, E	-	slowest
h	-	slow
ϕ, γ	-	fast
Autopilot Dynamics	-	fastest

3.5.1 Slowest Time Scale

$$\gamma_1 = \sigma_1 = 0$$

$$L_1 = mg \quad (3.73)$$

and

$$H_1 = \lambda_{x_1} V_1 \cos \phi_1 + \lambda_{y_1} V_1 \sin \phi_1 + \lambda_{E_1} V_1 (T-D)/mg + (1 - \xi) \quad (3.74)$$

λ_{x_1} and λ_{y_1} are constants. Optimal ϕ and h are obtained from

$$\frac{\partial H}{\partial \phi} = -\lambda_{x_1} V_1 \sin \phi_1 + \lambda_{y_1} V_1 \cos \phi_1 = 0 \Rightarrow \tan \phi_1 = \frac{\lambda_{y_1}}{\lambda_{x_1}} \quad (3.75)$$

$$\begin{aligned} \frac{\partial H}{\partial h} = & [\lambda_{x_1} \cos \phi_1 + \lambda_{y_1} \sin \phi_1 + \lambda_{E_1} (T-D)/mg] [-g/V_1] \\ & - \frac{\lambda_{E_1} V}{mg} \frac{\partial D}{\partial h} \end{aligned} \quad (3.76)$$

$$\text{Note that since } V_1 = \sqrt{2g(E-h)}, \quad \partial V_1 / \partial h = -g/V_1 \quad (3.77)$$

$$H(t_f) = 0$$

gives

$$\begin{aligned}
& (\lambda_{x_1} \cos \phi_1 + \lambda_{y_1} \sin \phi_1) V_1(t_f) + \frac{\xi V_1(t_f)}{E_0 m g} [T(t_f) - D(t_f)] \\
& + (1 - \xi) = 0
\end{aligned} \tag{3.78}$$

or

$$(\lambda_{x_1} \cos \phi_1 + \lambda_{y_1} \sin \phi_1) = - \frac{\xi [T(t_f) - D(t_f)]}{E_0 m g} - \frac{(1 - \xi)}{V_1(t_f)} \tag{3.79}$$

Eqs. (3.75) and (3.79) give the solutions for λ_{x_1} and λ_{y_1} .

Nominally, a differential equation must be solved to obtain E . Because the Hamiltonian is explicitly time dependent through T and m the Hamiltonian is not constant throughout. We will use an approximation which will avoid the need to solve the differential equation in the backward direction. The approximation consists of using average values of T and m in the definition of the Hamiltonian. Thus,

$$H_1 = \lambda_{x_1} V_1 \cos \phi_1 + \lambda_{y_1} V_1 \sin \phi_1 + \frac{\lambda_{E_1} V_1 (T_{av} - D_1)}{m_{av} g} + (1 - \xi) \tag{3.80}$$

With this approximation

$$H_1(t) = 0$$

Therefore,

$$\lambda_{E_1} \frac{(T_{av} - D_1)}{m_{av} g} + \frac{1 - \xi}{V_1} = \frac{\xi [T(t_f) - D_1(t_f)]}{m_{av} g E_0} + \frac{1 - \xi}{V_1(t_f)} \tag{3.81}$$

Using Eq. (3.80), Eq. (3.76) becomes

$$\frac{g}{V_1^2} (1 - \xi) - \frac{\lambda_{E_1} V_1}{m_{av} g} \frac{\partial D}{\partial h} = 0 \tag{3.82}$$

$$\frac{\xi}{V_1^2} (1-\xi)(T_{av}-D_1) = \left[\frac{\xi [T(t_f)-D(t_f)]}{m_{av}gE_0} + \frac{1-\xi}{V_1(t_f)} - \frac{1-\xi}{V_1} \right] V_1 \frac{\partial D}{\partial h} \quad (3.83)$$

This equation may be solved to obtain the optimal altitude. Note that the three adjoint variables are

$$\lambda_{x_1} = -\cos \phi_1 \left[\frac{\xi [T(t_f)-D_1(t_f)]}{E_0mg} + \frac{(1-\xi)}{V_1(t_f)} \right] \quad (3.84)$$

$$\lambda_{y_1} = -\sin \phi_1 \left[\frac{\xi [T(t_f)-D_1(t_f)]}{E_0mg} + \frac{(1-\xi)}{V_1(t_f)} \right] \quad (3.85)$$

$$\lambda_{E_1} = \frac{m_{av}g}{(T_{av}-D_1)} \left[\frac{\xi [T(t_f)-D_1(t_f)]}{E_0mg} + \frac{(1-\xi)}{V_1(t_f)} - \frac{(1-\xi)}{V_1} \right] \quad (3.86)$$

The two special cases corresponding to minimum time ($\xi = 0$) and minimum energy loss ($\xi = 1$) are shown in Table 3-2. For the minimum energy loss case, the drag is the lowest at the optimal altitude.

The flight path direction in the horizontal plane is determined by the horizontal projection of the line which joins the current missile position to the intended final missile position.

3.5.2 Altitude Dynamics

To solve for the altitude dynamics λ_{x_1} , λ_{y_1} , and λ_{E_1} are used from the previous time frame. Since flight path angle dynamics are still faster than altitude dynamics, setting and γ to zero, we get the following equilibrium conditions

$$\sigma_2 = 0 \quad (3.87)$$

$$L_2 = mg \cos \gamma_2 \quad (3.88)$$

Table 3-2 Optimal Solutions in the Slow Dynamics for the Minimum Time and the Minimum Energy Loss Problems

	Minimum Time ($\xi=0$)	Minimum Energy Loss ($\xi=1$)
Equation for Optimal Altitude	$\frac{g(T_{av}-D)}{V_1^2} = \frac{[V_1 - V_1(t_f)]}{V_1(t_f)} \frac{\partial D}{\partial h}$	$\frac{\partial D}{\partial h} = 0$
λ_{x1}	$-\frac{\cos \phi_1}{V_1(t_f)}$	$-\frac{\cos \phi_1 (T_{av}-D)}{E_0 mg}$
λ_{y1}	$-\frac{\sin \phi_1}{V_1(t_f)}$	$-\frac{\sin \phi_1 (T_{av}-D)}{E_0 mg}$
λ_{E1}	$\frac{m_{av} g^2}{V_1^3 \partial D / \partial h}$	Zero

where the subscript '2' denotes variables in the altitude dynamics. The corresponding Hamiltonian can be written as

$$H_2 = \lambda_{x_1} V_2 \cos \gamma_2 \cos \phi_1 + \lambda_{y_1} V_2 \cos \gamma_2 \sin \phi_1 + \lambda_{h_2} V_2 \sin \gamma_2 + \lambda_{E_1} V_2 (T - D_2) / mg + (1 - \xi) \quad (3.89)$$

The optimal flight path angle in the vertical plane is

$$\frac{\partial H_2}{\partial \gamma_2} = 0 = -(\lambda_{x_1} \cos \phi_1 + \lambda_{y_1} \sin \phi_1) V_2 \sin \gamma_2 + \lambda_{h_2} V_2 \cos \gamma_2 \quad (3.90)$$

Using Equation (3.26), we get

$$\tan \gamma_2 = \frac{\lambda_{h_2}}{(\lambda_{x_1} \cos \phi_1 + \lambda_{y_1} \sin \phi_1)} \quad (3.91)$$

Normally we would have to solve a differential equation to obtain λ_{h_2} because T and m are explicit time functions. However, we would again set T and m to their average values which gives an additional equation:

$$H_2(t) = 0$$

$$(\lambda_{x_1} \cos \phi_1 + \lambda_{y_1} \sin \phi_1) V_2 \cos \gamma_2 + \lambda_{h_2} V_2 \sin \gamma_2 + \lambda_{E_1} V_2 \frac{(T_{av} - D_2)}{m_{av} g} + (1 - \xi) = 0 \quad (3.92)$$

Using Eq. (3.91) to eliminate λ_{h_2} , we get

$$(\lambda_{x_1} \cos \phi_1 + \lambda_{y_1} \sin \phi_1) \sec \gamma_2 + \frac{\lambda_{E_1} (T_{av} - D_2)}{m_{av} g} + \frac{1 - \xi}{V_2} = 0 \quad (3.93)$$

Eq. (3.80) is rewritten as

$$(\lambda_{x_1} \cos \phi_1 + \lambda_{y_1} \sin \phi_1) + \frac{\lambda_{E_1} (T_{av} - D_1)}{m_{av} g} + \frac{1-\xi}{V_1} = 0 \quad (3.94)$$

Subtracting Eq. (3.94) from Eq. (3.93)

$$\begin{aligned} & (\lambda_{x_1} \cos \phi_1 + \lambda_{y_1} \sin \phi_1)(\sec \gamma_2 - 1) + \frac{\lambda_{E_1} (D_1 - D_2)}{m_{av} g} \\ & + \frac{(1-\xi)(V_1 - V_2)}{V_1 V_2} = 0 \end{aligned} \quad (3.95)$$

From Eq. (3.76) and Eq. (3.80) we can find one solution to λ_{E_1}

$$\frac{\lambda_{E_1}}{m_{av} g} = \frac{(1-\xi)g/V_1^2}{V_1 \partial D_1 / \partial h} \quad (3.96)$$

Therefore Eq. (3.95) is simplified to

$$\begin{aligned} & (\lambda_{x_1} \cos \phi_1 + \lambda_{y_1} \sin \phi_1)(\sec \gamma_2 - 1) = \\ & - (1-\xi) \left[\frac{V_1 - V_2}{V_1 V_2} + \frac{(D_1 - D_2)g}{V_1^3 \partial D_1 / \partial h} \right] \end{aligned} \quad (3.97)$$

This equation gives a value of $\sec \gamma_2$. γ_2 is positive if $h_2 < h_1$ and is negative if $h_1 < h_2$. Note that if this equation solves out to a negative value of $(\sec \gamma_2 - 1)$, the desired flight path angle is $\pm 90^\circ$ (this happens because Eq. (3.90) must be modified when optimal γ_2 is 90°). Note that $(\lambda_{x_1} \cos \phi_1 + \lambda_{y_1} \sin \phi_1)$ is obtained from Eq. (3.84) and (3.85)

$$\lambda_{x_1} \cos \phi_1 + \lambda_{y_1} \sin \phi_1 = - \left[\frac{\xi [T(t_f) - D_1(t_f)]}{E_0 m g} + \frac{(1-\xi)}{V_1(t_f)} \right] \quad (3.98)$$

3.5.3 Attitude Dynamics

Given desired values of the flight path angles and the current values of flight path angles, the lift vector and its orientation can be computed using techniques developed previously [8]. We shall summarize them here since they form an important part of the overall guidance law. The orientations of the current velocity vector and the desired velocity vector are

$$\vec{v}_c = [\cos \gamma \cos \phi, \cos \gamma \sin \phi, \sin \gamma] \quad (3.99)$$

$$\vec{v}_d = [\cos \gamma_2 \cos \phi_1, \cos \gamma_2 \sin \phi_1, \sin \gamma_2] \quad (3.100)$$

The total angle through which the flight path must be changed is given by

$$\Delta\psi = \arccos (\vec{v}_c \cdot \vec{v}_d) \quad (3.101)$$

The lift vector net of gravity must be perpendicular to \vec{v}_c , in the plane containing \vec{v}_c and \vec{v}_d . The value of the lift depends on $\Delta\psi$ and the variation of drag with lift. It was shown previously that the lift is of the form

$$L \approx K \sqrt{\Delta\psi} \quad (3.102)$$

K may depend on dynamic pressure and time-to-go. The value of K may be derived from the results of [8].

3.6 ALGORITHM

The algorithm involves the following steps:

1. Based on current missile state, compute γ , ϕ , density, drag and total energy.
2. Computer desired ϕ , based on the line joining the current missile location in the horizontal and the desired missile location at the end of the midcourse phase.

3. Based on the drag, compute an approximate value of $V_1(t_f)$, $D(t_f)$ and time-to-go. Time-to-go is used to determine T_{av} and m_{av} .
4. Solve Eq. (3.83) for the optimal altitude. Eqs. (3.84), (3.85) and (3.86) give λ_{x_1} , λ_{y_1} and λ_{E_1} .
5. Solve Eq. (3.97) for γ_2 . Note that for computational advantages $(\sec \gamma_2 - 1)$ should be approximated as

$$\frac{2 \sin^2(\gamma_2/2)}{\cos \gamma_2}$$

particularly when γ_2 is small.

6. Compute \vec{v}_c , \vec{v}_d and $\Delta\psi$ using Eqs. (3.99) to (3.101). A numerically desirable way to compute $\Delta\psi$ uses the formula

$$\Delta\psi = 2 \arcsin \left| \frac{1}{2}(\vec{v}_c - \vec{v}_d) \right|$$

The vertical bars represent the 2-norm of a matrix. Compute the desired lift using Eq. (3.102).

7. The orientation of the lift vector is given by V_c and V_d .
8. Add the component of gravity perpendicular to the velocity vector to the lift computed in steps (3.59) and (3.60).

SECTION 4

TARGET DYNAMICS ESTIMATION AND ADAPTIVE TRAJECTORY TRACKING

4.1 EQUATIONS OF MOTION

For state estimation, we will describe the relative dynamics of the target with respect to the missile in the inertial axis system. If \vec{x} is the vector of relative position, \vec{v}_t and \vec{v}_m are vectors of target and missile velocity, respectively, and \vec{a}_t and \vec{a}_m are the corresponding acceleration vectors, we have

$$\dot{\vec{x}} = \vec{v}_t - \vec{v}_m, \quad (4.1)$$

$$\dot{\vec{v}} = \vec{a}_t, \quad (4.2)$$

$$\dot{\vec{v}} = \vec{a}_m. \quad (4.3)$$

\vec{a}_m may be measured by on-board missile sensors. For state estimation to be used in the missile guidance law development, the target acceleration must be modeled. Behavior of target acceleration components is usually different along target longitudinal and lateral directions. To model these components separately, the targets's orientation angle in the inertial axis system must be estimated. Since it is difficult to estimate target orientation angles, it is desirable to model the target acceleration components in the same manner in all directions. One often-used formulation is a random walk model:

$$\dot{\vec{a}}_t = \eta, \quad (4.4)$$

where η is a vector of white Gaussian noise sources. Note

that the missile target dynamic model in the inertial axis system is linear.

Nonlinearities in the state estimator arise from the measurements. Passive seekers measure pitch and yaw gimbal angles θ_p and θ_y with respect to the missile body frame (m refers to quantities in the missile frame)

$$\theta_p = \arctan (z_m/x_m) , \quad (4.5)$$

$$\theta_y = \arctan \left(y_m / \sqrt{x_m^2 + z_m^2} \right) . \quad (4.6)$$

The inertial line-of-sight may also be considered as potential measurements. The range measurement is

$$R = (x_m^2 + y_m^2 + z_m^2)^{1/2} . \quad (4.7)$$

Since the measurement model is nonlinear, an extended Kalman filter could be used to estimate states. The covariance equations need to be propagated in parallel with the estimated state equations. Assuming that the missile velocity components may be determined by open-loop integration of missile acceleration components, a nine-state Kalman filter will be required to estimate relative position, target velocity and target acceleration. The covariance equations will place significant computation requirements on the on-line processor. The time-scale separation approach will attempt to simplify this problem.

4.2 TIME-SCALE SEPARATION FOR STATE ESTIMATION

In the state estimator, the time-scale separation methods will consider the position as the slowest states, followed by target velocity and target acceleration.

Since the measurements are all in the slowest time frame, we will attempt estimation in that time frame first. The estimator equations should be of the form (the superscript '^')

denotes estimate of a quantity):

$$\hat{\dot{x}} = \hat{\dot{v}}_t - \hat{\dot{v}}_m + K_x(y - h(\hat{x})) , \quad (4.8)$$

where the measurements are represented by

$$y = h(\hat{x}) . \quad (4.9)$$

The gain K_x is computed assuming $\hat{\dot{v}}_t$ and $\hat{\dot{v}}_m$ do not change significantly during this time period.

The variables in the next faster time scale will be estimated by considering a measurement of \dot{v}_t based on the position equation. One possible model for pseudo-velocity measurement is

$$\begin{aligned} y_v &= \hat{\dot{x}} + \hat{\dot{v}}_m \\ &= K_x(y - h(\hat{x})) + \hat{\dot{v}}_t . \end{aligned} \quad (4.10)$$

The estimator for target velocity components then becomes

$$\begin{aligned} \hat{\dot{v}}_t &= \hat{\dot{a}}_t + K_v(y_v - \hat{\dot{v}}_t) \\ &= \hat{\dot{a}}_t + K_v K_x(y - h(\hat{x})) . \end{aligned} \quad (4.11)$$

In the missile guidance problem, noisy measurements of the target position components are available to the missile in the form of look angles and possibly range. From these measurements the trajectory of the target is to be estimated. Further, this trajectory must be updated with each new measurement. It appears feasible to represent the target acceleration as an auto-regressive (AR) model. Under the assumption that the measurement noise has a rational spectrum, it can be shown that the corresponding model to be fitted to the noisy acceleration samples, derived from the position measurements, is an auto-regressive moving-

average (ARMA) model whose denominator coefficients are the desired parameters. These parameters can be estimated using recursive maximum likelihood (RML) or recursive prediction error (RPE) algorithms, which are described in Section 4.3. Once the model parameters have been estimated, the target position can be predicted in one of two ways: 1) Estimate the acceleration from the derived model and determine the position estimate by integrating twice, and 2) The target position satisfies an AR model whose AR coefficients can be computed from the above estimated coefficients. The target position can then be predicted using these AR coefficients.

The estimated target state variables serve as input to the algorithm of Section 3, which computes the optimal missile state variables. These nominal values are used by an adaptive autopilot as a reference; the autopilot provides actual control signals to the missile control surfaces, in order to track the reference missile trajectory obtained from the optimal control solution. A recursive lattice algorithm for implementing the adaptive autopilot is described in Section 4.4.

4.3 RECURSIVE MAXIMUM LIKELIHOOD (RML) ARMA IDENTIFICATION

Let the observed time series be modeled as

$$A(q^{-1}) y_k = C(q^{-1}) e_k \quad (4.12)$$

where

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \dots + a_L q^{-L} \\ C(q^{-1}) &= 1 + c_1 q^{-1} + \dots + c_N q^{-N} \end{aligned} \quad (4.13)$$

with q^{-1} as the delay operator

$$q^{-1} y_k = y_{k-1} .$$

The recursive maximum likelihood (RML) algorithm computes an estimate $\hat{\theta}$ of the model parameter vector θ

$$\theta^T = (a_1 \dots a_L \quad c_1 \dots c_L) \quad (4.14)$$

by the following set of recursions:

$$\hat{\theta}_k = \hat{\theta}_{k-1} + P_k \psi_k \epsilon_k \quad (4.15)$$

$$P_k = P_{k-1} - \frac{P_{k-1} \psi_k \psi_k^T P_{k-1}}{1 + \psi_k^T \psi_{k-1} \quad k}$$

where

$$\epsilon_k := y_k - \hat{\theta}_{k-1}^T y_k \quad (4.16)$$

$$y_k^T := [-y_{k-1} \dots -y_{k-L} \quad \epsilon_{k-1} \dots \epsilon_{k-N}]$$

$$\psi_k := y_k / C(q^{-1})$$

The Recursive Prediction Error (RPE) algorithm uses modified ϵ , y , ψ variables. The prediction error ϵ_k is replaced by the a posteriori residual

$$\bar{\epsilon}_k := y_k - \hat{\theta}_k^T \bar{y}_k \quad (4.17)$$

where

$$\bar{y}_k := [-y_{k-1} \dots -y_{k-L} \quad \bar{\epsilon}_{k-1} \dots \bar{\epsilon}_{k-N}] \quad (4.18)$$

The variable ψ_k is also modified to

$$\bar{\psi}_k := [-y_{k-1} \dots -y_{k-L}, \quad \tilde{\epsilon}_{k-1} \dots \tilde{\epsilon}_{k-L}] \quad (4.19)$$

where

$$\tilde{\epsilon}_k = \epsilon_k / C(q^{-1}) \quad (4.20)$$

A detailed discussion of both algorithms is provided in Appendix B.

4.4 ADAPTIVE LATTICE ALGORITHM FOR TRAJECTORY TRACKING

The missile trajectory has to be adaptively controlled by the autopilot to track the nominal trajectory, which is itself adaptively updated (at a lower rate) according to the changes in target trajectory estimates. An algorithm to compute the control inputs that achieve this objective is described in Appendix D. The main part of the algorithm (everything except step V in the Appendix) is devoted to establish the exact input-output relationship of the missile and to update this relationship as the missile parameters change. The lattice algorithm described in Appendix D consists of a cascade connection of identical lattice modules (Figure 4-1). The matrix computations performed by each section can be transformed into a set of inter-related scalar recursions, so that each multichannel module of Figure 4-1 is replaced by a square array of single-channel lattice cells (Figure 4-2). Each of these cells performs a set of scalar computations summarized in Table 4-1. The modular architecture of the algorithm not only makes it perfectly suitable for VLSI implementation but also provides it with a better numerical behavior and higher throughput rate than any direct implementation of the recursions of Appendix D on a general purpose computer.

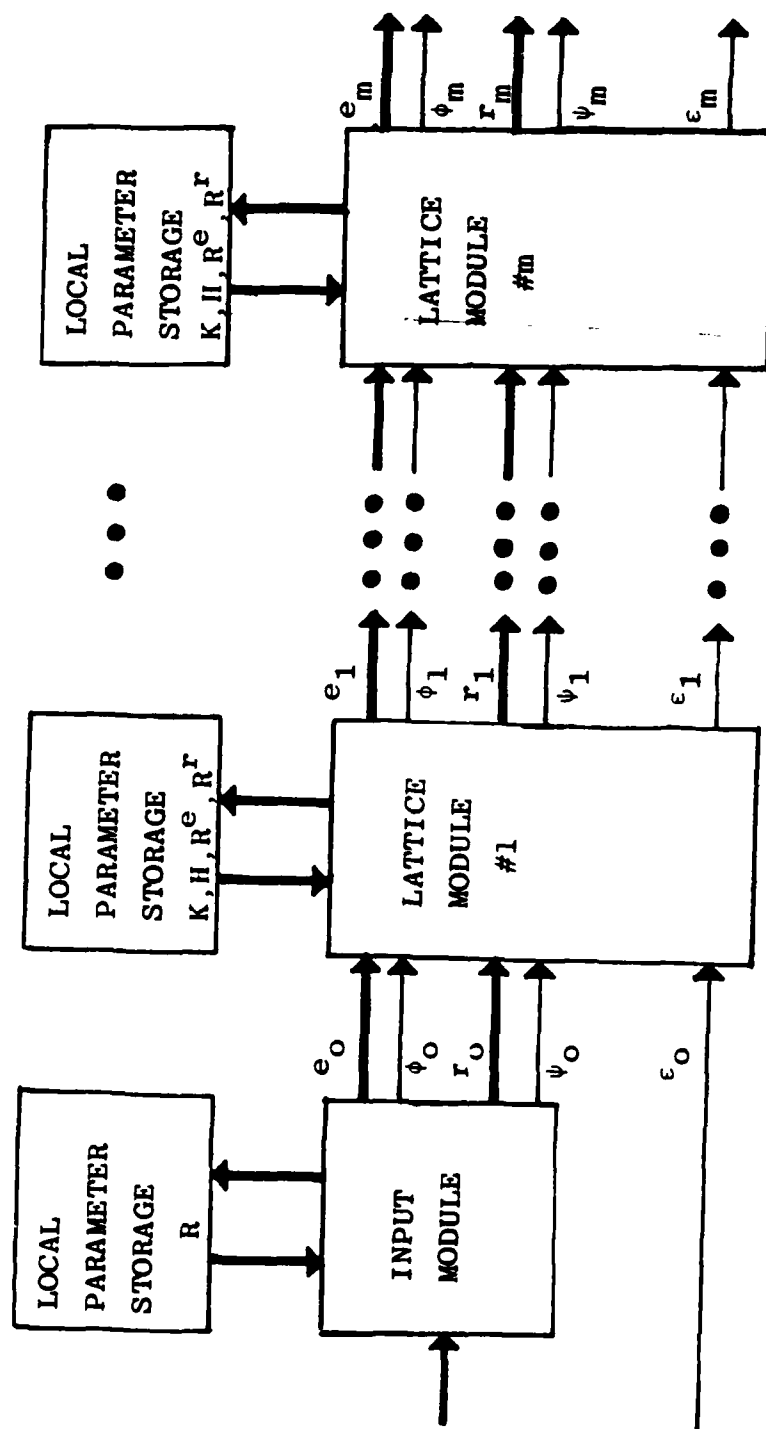


Figure 4-1 Architecture of the Adaptive Multichannel
Lattice Algorithm for Trajectory Tracking

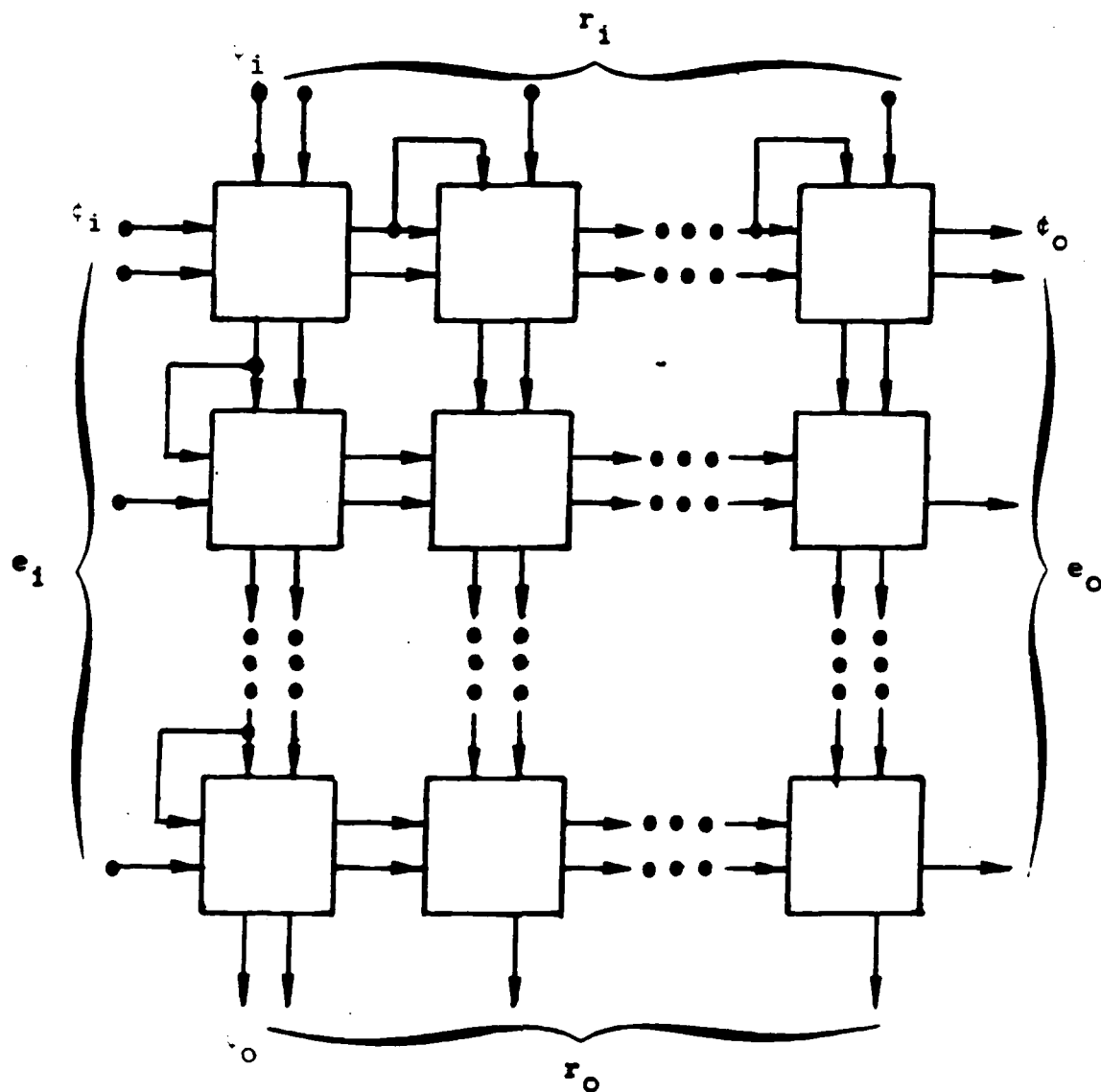


Figure 4-2 Modular Architecture of a Multichannel Lattice Section
 e - forward residuals
 r - backward residuals
 ϕ, ψ - auxilliary signal

Table 4-1 Recursions for a Single-Channel Lattice Cell

$K = \lambda K_S + \frac{e_i r_i^*}{\phi_i}$ $R^e = \lambda R_S^e + \frac{e_i e_i^*}{\phi_i}$ $R^r = \lambda R_S^r + \frac{r_i r_i^*}{\psi_i}$
$e_o = e_i - K_S R_S^{-r} r_i$ $r_o = r_i - K_S^* R_S^{-e} e_i$
$\phi_o = \phi_i + \lambda^{-1} r_i^* R_S^{-r} r_i$ $\psi_o = \psi_i + \lambda^{-1} e_i R_S^{-e} e_i$
$H = \lambda H_S + \epsilon_i r_i^*$ $\epsilon_o = \epsilon_i - H_S R_S^{-r} r_i$

SECTION 5

SUMMARY

This section summarizes the work performed under this research effort. Plans for future work are also presented.

5.1 ADVANCED GUIDANCE LAWS

Proportional navigation (pronav) guidance, which has been highly successful in the past, is not likely to meet future missile guidance requirements; for example, in an anti-tactical ballistic missile (ATBM) scenario. A singular perturbation approximation to modern control theory provides much high accuracy than pronav. The mathematical formulation of the solution is significantly simplified by introducing separation of time scales. The resulting algorithm is well suited for on-board real-time implementation.

5.2 ADAPTIVE TARGET STATE ESTIMATION AND TRACKING

Efficient and fast algorithms for adaptive target state estimation and tracking have been developed. The recursive maximum likelihood (RML) algorithm for fitting an auto-regressive moving-average (ARMA) model to noisy target position measurements exhibits fast convergence and is unbiased in the presence of measurement noise. The adaptive multichannel lattice algorithm for trajectory tracking has excellent numerical behavior, fast convergence and a modular structure that makes it perfectly suitable for parallel processing implementation.

5.3 FUTURE RESEARCH

The following areas need further study:

1. Real-time implementation of the advanced guidance law algorithm and evaluation on complete ATBM simulation,
2. Application of the recursive maximum likelihood (RML) algorithm to target trajectory estimation,
3. Application of the recursive adaptive lattice-form controller to design a robust adaptive autopilot for medium-range surface-to-air missiles, and
4. Architectures for an integrated, parallel, multi-microprocessor implementation of the missile guidance and control system.

REFERENCES

- [1] "Advanced Analysis for Future Missiles, " Guidance and Control Directorate, U.S. Army Missile Command, Redstone Arsenal, Alabama, November 1979.
- [2] Gupta, N. K., and C. Z. Gregory, "Advanced Guidance Laws for Advanced Aircraft Missiles in Medium-Range Intercepts," ISI Technical Memo 5032-01, prepared for Army Missile Command, December 1981.
- [3] Gupta, N. K., "Advanced Guidance Laws for Army Tactical Missiles," ISI-02, prepared for the U.S. Army Missile Command, October, 1980.
- [4] Reddy, V. U., R. H. Travassos, and T. Kailath, "A Comparison of Nonlinear Spectral Estimation Techniques," ISI Technical Memo 5016-05, January 1982.
- [5] Reddy, V. U., "A Numerically Stable and High-Speed Recursive Least-Squares Ladder-Form Algorithm," ISI Technical Memo, ISI 12, November 1981.
- [6] Lev-Ari, H., "Wave-Front Array Processing Architectures for Multichannel Ladder Algorithms," ISI Technical Memo 5016-07, June 1982.
- [7] Shah, S. C., "Adaptive Control and Prediction Using Lattice Structures," ISI Technical Memo 5017-01, October 1981.
- [8] Sridhar, B., and N. K. Gupta, "Missile Guidance Laws Based on Singular Perturbation Methodology," J. Guidance and Control, Vol. 3, No. 2, April 1980, pp. 158-165.

APPENDIX A

PROPORTIONAL NAVIGATION GUIDANCE

1. EQUATIONS

With reference to Figure 1, the engagement dynamics in two dimensions are [2]

$$\dot{r}\dot{\sigma} = v_t \sin \phi_t - v_m \sin \phi_m, \quad \sigma(t_0) = \sigma_0, \quad (1)$$

$$\dot{r} = v_t \cos \phi_t - v_m \cos \phi_m, \quad r(t_0) = r_0. \quad (2)$$

By differentiating the first equation, adding $\dot{\sigma}$ times the second equation to it, we get

$$r\ddot{\sigma} + 2\dot{r}\dot{\sigma} + \cos \phi_m a_m = -\dot{v}_m \sin \phi_m + v_t \cos \phi_t \dot{\theta}_t + v_t \sin \phi_t \dot{\sigma},$$

$$\dot{\sigma}(t_0) = \sigma_0. \quad (3)$$

Note that $a_m = v_m \dot{\theta}_m$. In this section, we consider missiles with constant speed, and targets with no lateral acceleration capabilities (significant further deterioration occurs if these assumptions do not hold). The change in target speed is included as follows:

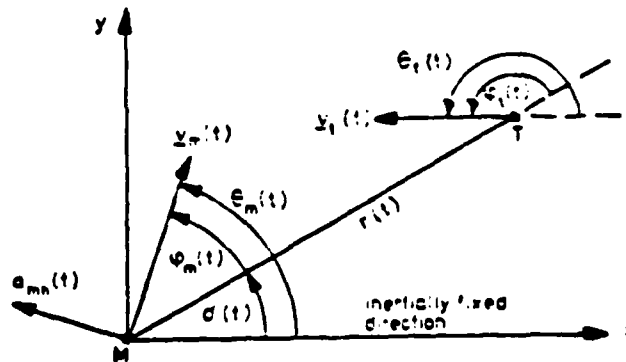


Figure 1. Definitions of the Kinematic Variables for the Relative Motion Between the Missile M and the Target T

$$r\ddot{\sigma} + 2\dot{r}\dot{\sigma} + \cos \phi_m a_m = \dot{v}_t \sin \phi_t . \quad (4)$$

The commanded acceleration for pronav guidance law is of the form

$$a_{mc} = -\Lambda_n \frac{r\dot{\sigma}}{\cos \phi_m} , \quad (5)$$

Λ_n is the navigation gain. If the missile dynamics is negligible, i.e., $a_m = a_{mc}$, the closed-loop dynamics is

$$r\ddot{\sigma} + (2 - \Lambda_n)\dot{r}\dot{\sigma} = \dot{v}_t \sin \phi_t ,$$

which is stable as long as $\Lambda_n > 2$ (note that \dot{r} is negative).

Missile dynamics, approximated by a second-order system, can cause instabilities

$$\ddot{a}_m + 2\xi_m \omega_m \dot{a}_m + \omega_m^2 a_m = \omega_m^2 a_{mc} . \quad (6)$$

A pronav guidance schematic flowchart is shown in Figure 2. The characteristic polynomial of the closed-loop system is

$$s^3 + \left(\frac{2\dot{r}}{r} + 2\xi_m \omega_m\right)s^2 + \left(\frac{4\dot{r}}{r} \xi_m \omega_m + \omega_m^2\right)s + \frac{\dot{r}}{r} \omega_m^2 (2 - \Lambda_n) . \quad (7)$$

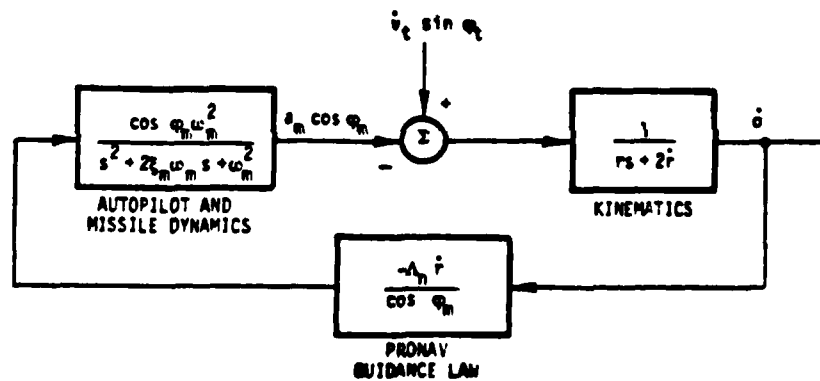


Figure 2. Pronav Guidance Loop

2. ANALYSIS OF STABILITY

The guidance law is stable, i.e., the polynomial of Eq. 7 has roots with negative real parts, if

$$\begin{aligned}\Lambda_n &> 2, \\ -\frac{\ddot{r}}{\dot{r}} &> \frac{1}{\xi_m \omega_m}, \\ -\frac{\ddot{r}}{\dot{r}} &> \frac{4\xi_m}{\omega_m},\end{aligned}$$

and using the Routh Hurwitz criteria

$$-\frac{\ddot{r}}{\dot{r}} > 2\left(a + \sqrt{a^2 - 1}\right)/\omega_m,$$

where

$$a = \xi_m + \frac{\Lambda}{8\xi_m}. \quad (8)$$

Note that \dot{r} is nominally negative and $-r/\dot{r}$ is the time-to-go if there are no maneuvers. If Λ_n exceeds two, the second and the third condition are always less restrictive than the last condition. Therefore, conditions for stability are

$$\begin{aligned}\Lambda_n &> 2, \\ t_{go} &> 2\left(a + \sqrt{a^2 - 1}\right)/\omega_m.\end{aligned} \quad (9)$$

Note that the pronav drives the missile unstable prior to impact for all values of damping ratio. A root locus plot with Λ_n is shown in Figure 3 to illustrate the stability problem. Note that $\Lambda_n > 2$ to stabilize the kinematic pole. The point where the complex pole pair crosses the imaginary axis depends on time-to-go. As the time-to-go decreases, this crossing occurs at smaller and smaller Λ_n .

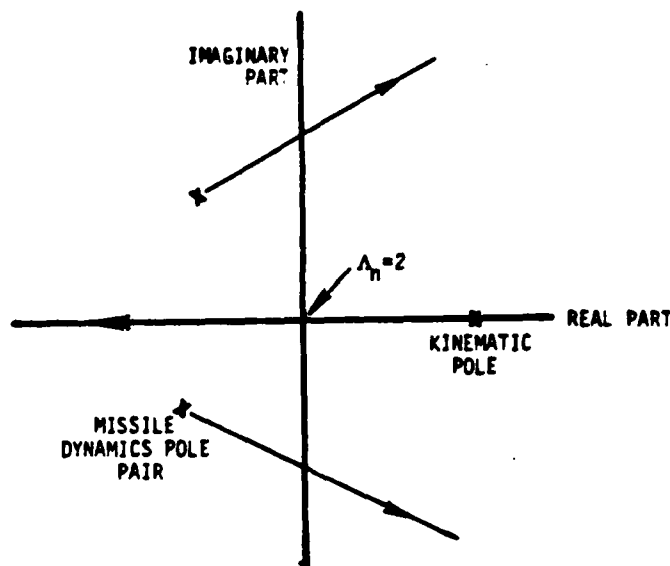


Figure 3. Root Locus with Λ_n

Figure 4 shows stability regions for various pronav gains and two values of missile damping. The higher the pronav gain, the earlier the missile goes unstable.

3. SUMMARY

When the target has high acceleration components the terminal instability of pronav causes large miss distances. The pronav is also too slow to respond to large errors when the range is large, causing missile saturation prior to impact.

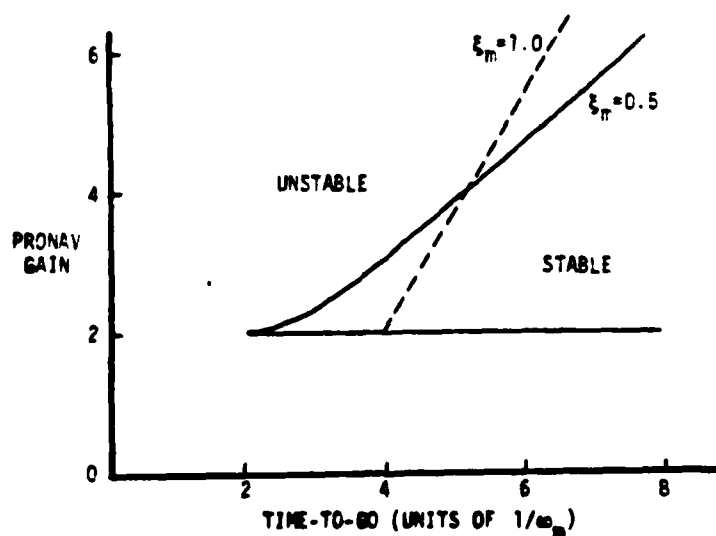


Figure 4. Regions of Missile Stability

APPENDIX B

A QUASI-NEWTON ALGORITHM FOR ML ESTIMATION
OF ARMA MODEL PARAMETERS

Let the observed time series $\{y(t), t = 0, 1, \dots\}$ be modelled as

$$A(q^{-1}) y(t) = C(q^{-1}) e(t) \quad (1)$$

where

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \dots + a_L q^{-L} \\ C(q^{-1}) &= 1 + c_1 q^{-1} + \dots + c_N q^{-N} \end{aligned} \quad (2)$$

with q^{-1} as the delay operator

$$q^{-1} y(t) = y(t-1),$$

and $e(t)$ is a sequence of zero mean white noise samples.
Rewriting (1) as

$$y(t) = \left[1 - \frac{A(q^{-1})}{C(q^{-1})} \right] y(t) + e(t) \quad (3)$$

the one-step ahead predictor $\hat{y}(t|\theta)$ is given by

$$\hat{y}(t|\theta) = \left[1 - \frac{A(q^{-1})}{C(q^{-1})} \right] y(t) \quad (4)$$

where θ is the parameter vector as defined in (6).

From (3) and (4), the prediction error $\varepsilon(t, \theta)$ is

$$\varepsilon(t, \theta) = y(t) - \hat{y}(t|\theta) = y(t) \theta^T \zeta(t, \theta) \quad (5)$$

where

$$\theta^T = (a_1 \dots a_L \ c_1 \dots c_N) \quad (6)$$

$$\zeta^T(t, \theta) = (-y(t-1) \dots -y(t-L) \ \varepsilon(t-1, \theta) \dots \varepsilon(t-N, \theta))$$

The off-line maximum likelihood (ML) method of estimating the parameter vector θ corresponds to minimizing the function

$$J = \frac{1}{2} \sum_{s=0}^t \varepsilon^2(s, \theta) \quad (7)$$

The above minimization problem is non-linear in θ and hence an explicit solution is not possible. Therefore, a numerical search procedure based on a Quasi-Newton method will be used to find θ .

Differentiating J with respect to θ gives

$$\nabla J = \sum_{s=0}^t \varepsilon(s, \theta) \nabla \varepsilon(s, \theta) \quad (8)$$

where ∇J and $\nabla \varepsilon(s, \theta)$ denote the gradient vector of J and $\varepsilon(s, \theta)$, respectively, with respect to θ . Differentiating once again gives

$$\nabla^2 J = \sum_{s=0}^t [\nabla \varepsilon(s, \theta) \nabla^T \varepsilon(s, \theta) + \varepsilon(s, \theta) \nabla^2 \varepsilon(s, \theta)] \quad (9)$$

From the expression (5),

$$\nabla \varepsilon(s, \theta) = -\nabla \hat{y}(s|\theta) = -\psi(s, \theta) \quad (10)$$

where $\psi(s, \theta)$ denotes the gradient vector of the prediction $\hat{y}(s, \theta)$.

The Hessian, $\nabla^2 J$, will now be approximated. At the true minimum of (8), the second term of (9) can be shown to be zero. Since the true Hessian is more important close to the true minimum than elsewhere, we approximate the Hessian by the first term of (9) which, in view of (10), becomes

$$\nabla^2 J \approx \sum_{s=0}^t \psi(s, \theta) \psi^T(s, \theta) \quad (11)$$

Further approximations are needed in order to obtain a recursive Quasi-Newton algorithm from (8) and (11). Computation of $\epsilon(t, \theta)$ requires all the data up to t . This computation is approximated by using the latest values of the data and the parameter estimates, and denote the corresponding $\zeta(t, \theta)$, $\psi(t, \theta)$ and $\epsilon(t, \theta)$ by $\zeta(t)$, $\psi(t)$ and $\epsilon(t)$, respectively. The Quasi-Newton update of the parameter vector is then given by:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \left[\sum_{s=0}^t \psi(s) \psi^T(s) \right]^{-1} \psi(t) \epsilon(t) \quad (12)$$

where we approximated ∇J by $\psi(t)\epsilon(t)$. Note that the effect of the above approximations on the asymptotic values of the parameter estimates is negligible if the roots of $C(z)$ lie inside the unit circle.

Now consider the computation of the gradient vector of the prediction. From (4),

$$\hat{y}(t) = \left[1 - \frac{A(q^{-1})}{C(q^{-1})} \right] y(t)$$

which gives

$$\frac{\partial}{\partial a_1} \hat{y}(t) = - \frac{y(t-1)}{C(q^{-1})} = - \hat{y}(t-1) \quad (13)$$

and

$$\frac{\partial \hat{y}(t)}{\partial c_i} = \frac{\varepsilon(t-i)}{C(q^{-1})} = \tilde{\varepsilon}(t-i) \quad (14)$$

Combining (3) and (14), we obtain

$$\nabla \hat{y}(t) = \psi(t) = \frac{\zeta(t)}{C(q^{-1})} \quad (15)$$

$\tilde{y}(t)$ and $\tilde{\varepsilon}(t)$ are the filtered variables. The vector $\psi(t)$ can be defined in terms of these variables as

$$\psi^T(t) = [-\tilde{y}(t-1) \dots -\tilde{y}(t-L) \tilde{\varepsilon}(t-1) \dots \tilde{\varepsilon}(t-N)] \quad (16)$$

In computing $\psi(t)$, the estimated value of θ at $t-1$, $\hat{\theta}(t-1)$, is used. It is then easy to see that

$$\tilde{y}(t) = y(t) - \hat{c}_1(t)\tilde{y}(t-1) \dots - \hat{c}_N(t)\tilde{y}(t-N) \quad (17)$$

$$\tilde{\varepsilon}(t) = \varepsilon(t) - \hat{c}_1(t)\tilde{\varepsilon}(t-1) \dots - \hat{c}_N(t)\tilde{\varepsilon}(t-N) \quad (18)$$

Using the matrix-inversion lemma, a recursive version for (2) can be obtained. The resulting algorithm follows:

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + P(t) \psi(t) \varepsilon(t) \\ P(t) &= P(t-1) - \frac{P(t-1) \psi(t) \psi^T(t) P(t-1)}{1 + \psi^T(t) P(t-1) \psi(t)} \end{aligned} \quad (19)$$

$$\varepsilon(t) = y(t) - \hat{\theta}^T(t-1) \zeta(t)$$

$$\hat{\theta}^T(t) = (\hat{a}_1(t) \dots \hat{a}_L(t) \hat{c}_1(t) \dots \hat{c}_N(t))$$

$\zeta(t)$ and $\psi(t)$ are as defined in (6) and (16), respectively. This algorithm is called the recursive maximum likelihood (RML) method.

Note here that the latest error term contained in $\zeta(t)$ is $e(t-1)$, and the latest filtered error in $\psi(t)$ is $\tilde{e}(t-1)$. Since the estimate $\hat{\theta}(t-1)$, available at the beginning of the t -th sampling interval, facilitates the computation of the a posteriori residual $\bar{e}(t-1)$, defined as

$$\bar{e}(t-1) = y(t-1) - \hat{\theta}^T(t-1) \zeta(t-1), \quad (20)$$

the residuals can be used in place of the prediction errors in $\zeta(t)$ and $\psi(t)$. Thus, the modified forms of $\zeta(t)$ and $\psi(t)$ are given by

$$\begin{aligned} \zeta^T(t) &= (-y(t-1) \dots -y(t-L) \bar{e}(t-1) \dots \bar{e}(t-N)) \\ \psi^T(t) &= (-y(t-1) \dots -y(t-L) \tilde{e}(t-1) \dots \tilde{e}(t-N)) \end{aligned} \quad (21)$$

where $\tilde{e}(t) = \bar{e}(t)/C(q^{-1})$. The algorithm (19) with $\zeta(t)$ and $\psi(t)$ as defined in (21) is called the recursive prediction error method (RPEM). The difference between (6), (16), and (21) reflects in the transient behavior of the RML method and the RPEM; however, their asymptotic behavior is identical.

APPENDIX C

THE ADAPTIVE LEAST-SQUARES PROBLEM

Exact-least-squares lattice algorithms provide recursive solutions to the following adaptive least squares problem:

Given two sequences of multichannel measurements

$$y(0), y(1), \dots$$

and

$$x(0), x(1), \dots$$

find a linear estimate of $x(k)$ based on m previous measurements of y , namely

$$\hat{x}(k) := \sum_{i=1}^m h_i y(k-i)$$

such that the exponentially weighted cost function

$$C_t^{m,\lambda} := \sum_{k=0}^t \lambda^{t-k} ||x(k) - \hat{x}(k)||^2$$

is minimized.

The optimal solution to this problem, is a function of m, λ, t and also of the data $y(k), x(k)$. Since it is customary in adaptive applications to solve the problem for several values of m and t , the dependence of the solution upon these parameters will be indicated explicitly by introducing the notation

$$h_{i,t}^m$$

for the solution that minimizes $C_t^{m,\lambda}$. The exponential weight is usually chosen in the range $0.9 \leq \lambda \leq 1.0$.

Adaptive lattice algorithms provide solutions which are recursive both in the parameter t ("time") and in the parameter m ("order"). These algorithms are based, consequently, on two update formulas: an order-update, which relates the solution $h_{i,t}^{m-1}$ to $h_{i,t}^m$, and a time-update, which relates $h_{i,t-1}^m$ to $h_{i,t}^m$. These update formulas can be expressed either in terms of the residuals

$$\epsilon_m(t) := x(t) - \sum_{i=1}^m h_{i,t}^m y(t-i) \quad (1)$$

or in terms of the prediction-errors

$$\epsilon_m^p(t) := x(t) - \sum_{i=1}^m h_{i,t-1}^m y(t-i) \quad (2)$$

The derivation of the time- and order-update formulas is based upon a geometric approach which is described in the sequel.

Define the rectangular matrices

$$\begin{aligned} y_t &:= [y(t), y(t-1), \dots, y(1), y(0), 0 \dots 0] \\ x_t &:= [x(t), x(t-1), \dots, x(1), x(0), 0 \dots 0] \end{aligned} \quad (3)$$

The number of rows in each matrix is determined by the number of channels of the corresponding signal. The length of each row (say N) is chosen large enough to guarantee the appearance of zero columns of the end of x_t, y_t for every t in consideration. The rows of x_t, y_t are therefore elements (vectors) in the linear space of row vectors of length N . Defining an inner product for every two row vectors of length N ,

$$\begin{aligned} \langle a, b \rangle &:= a \Lambda b^* \\ \Lambda &:= \text{diag} \{ \lambda^i; 0 \leq i \leq N-1 \} \end{aligned} \quad (4)$$

where the asterisk $*$ denotes the Hermitian transpose, we obtain a (weighted) Euclidean space. A collection of several vectors in this space, say x_1, x_2, \dots, x_k , written formally as a column

$$X := \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{pmatrix} \quad (5)$$

will be called a vector array (VA). The (inner) product of two VAs X, Y is defined as the matrix

$$\langle X, Y \rangle := [\langle x_i, y_j \rangle] \quad (6)$$

whose (i, j) element is $\langle x_i, y_j \rangle$, the inner product of the i -th vector in X with the j -th vector in Y .

The cost function $C_t^{m, \lambda}$ can be expressed in terms of the VAs x_t, y_t as

$$C_t^{m, \lambda} = \text{tr} \langle x_t - \sum_{i=1}^m h_i y_{t-i}, x_t - \sum_{i=1}^m h_i y_{t-i} \rangle$$

The solution $h_{i,t}^m$ that minimizes the cost function is obtained, by projecting the VA x_t (i.e., projecting every vector in x_t) on the subspace spanned by all the vectors contained in the VAs y_{t-t}, \dots, y_{t-m} . Since projections play a central role in solving the least squares problem, it will be convenient to introduce a shorthand notation. Let x_u denote the residual obtained by subtracting from every vector x_i in the VA X , the projection of x_i on the subspace spanned by the vectors contained in U . Then the minimal cost can therefore be expressed as

$$\begin{aligned} \min C_t^{m, \lambda} &= \text{tr} \langle \epsilon_{m,t}, \epsilon_{m,t} \rangle \\ \epsilon_{m,t} &:= (x_t)_{(y_{t-1}, \dots, y_{t-m})} = x_t - \sum_{i=1}^m h_{i,t}^m y_{t-i} \end{aligned} \quad (7)$$

where $\epsilon_{m,t}$ is the residual of x_t after removing its projection on $\text{span} \{y_{t-1}, \dots, y_{t-m}\}$.

The order- and time-update formulas of exact-least-squares lattice algorithms involve only the first column of the matrix

$\epsilon_{m,t}$; namely,

$$\epsilon_m(t) = \langle \epsilon_{m,t}, \pi \rangle \quad (8)$$

where

$$\pi := [1, 0, \dots, 0] \quad (9)$$

Notice that $\epsilon_m(t)$ is precisely the multichannel residual defined by (1). The prediction-error $\epsilon_m^p(t)$, defined by (2), is, similarly, the first column of the matrix

$$\epsilon_{m,t}^p := x_t - \sum_{i=1}^m h_{i,t-1}^m y_{t-i} \quad (10)$$

which seems to have no geometric interpretation in terms of projections. If the columns of this matrix are shifted one step to the left, and a column of zeros is introduced at the extreme right, the resulting matrix is

$$x_{t-1} - \sum_{i=1}^m h_{i,t-1}^m y_{t-1-i} = \epsilon_{m,t-1}$$

which is, of course, the residual of projecting x_t on span $\{y_{t-2}, \dots, y_{t-m-1}\}$. Denoting the left-shift on row vectors by D we have

$$\begin{aligned} Dy_t &= y_{t-1} \\ Dx_t &= x_{t-1} \end{aligned} \quad (11)$$

and as a consequence of our last argument

$$\begin{aligned} \epsilon_m^p(t) &= \langle \epsilon_{m,t}^p, \pi \rangle \\ D\epsilon_{m,t}^p &= \epsilon_{m,t-1} \end{aligned} \quad (12)$$

This identity can be used to derive order- and time-update formulas in terms of prediction-errors rather than residuals.

APPENDIX D

ADAPTIVE CONTROL AND PREDICTION USING LATTICE STRUCTURES

A large number of adaptive control and prediction algorithms can be described exactly as a lattice form implementation that requires only $O(M)$ computations per time update for an M^{th} order model. The well-known advantages of lattice form implementation in terms of numerical stability and simultaneous availability of lower order models become available for algorithms with known, desirable asymptotic convergence properties.

For the ARMAX (Autoregressive Moving Average with Exogenous Inputs) models, current state-of-the-art adaptive prediction either risks convergence to a local maxima of the likelihood function, or requires a Strict Positive Real condition for convergence. Over-parameterization of the predictor can be used to ensure that neither of these two problems would arise nor would the convergence rate reduce substantially. This is attractive only because of the $O(M)$ computational complexity of the algorithm.

INTRODUCTION

Consider a linear system with p inputs, p outputs described by:

$$A(q^{-1})y_t = q^{-1}B(q^{-1})u_t + C(q^{-1})v_t \quad (I)$$

q^{-1} = delay operator in discrete-time. $\{y_t\}$, $\{u_t\}$, and $\{v_t\}$ denote, respectively, the output, the input and the noise process, each of dimension $p \times 1$.

$$A(q^{-1}) = I + A_1q^{-1} + \dots + A_nq^{-n}$$

$$C(q^{-1}) = I + C_1q^{-1} + \dots + C_nq^{-n}$$

$$B(q^{-1}) = B_0 + B_1q^{-1} + \dots + B_mq^{-m}$$

$$\det(B_0) \neq 0.$$

Objective of the adaptive controller is to make the output follow a reference signal y_{ref_t} in the minimum-variance sense. Viz.

$$\text{Min } E\{|y_{t+1} - y_{ref_{t+1}}|^2 | y_t, u_t, y_{ref_{t+1}}, y_{t-1}, u_{t-1}, y_{ref_t}, \dots\}$$

Equation (1) can be rewritten in its prediction error form as

$$\begin{aligned} C(q^{-1})\{(y_{t+1} - y_{ref_{t+1}}) - v_{t+1}\} &= \{B(q^{-1})u_t + \\ &+ (C(q^{-1}) - A(q^{-1})) \cdot (y_t - y_{ref_t}) - A(q^{-1})y_{ref_{t+1}}\} \\ &= \theta_0 \phi_t - y_{ref_{t+1}} \end{aligned}$$

where

$$\begin{aligned} \phi_t &= [u_t^T, u_{t-1}^T, \dots, (y_{t-t} - y_{ref_t})^T, (y_{t-1} - y_{ref_{t-1}})^T, \dots, \\ &\quad y_{ref_t}^T, y_{ref_{t-1}}^T, \dots]^T \end{aligned}$$

Let $r = \max((n-1), m)$, $\dim \phi_t = 3 \cdot (r+1) \cdot p \times 1$ and θ_0 is a matrix of coefficients with dimension $= p \times \dim \phi_t$. θ_0 may have zero elements. Model order $M = r + 1$.

$$\begin{aligned} \theta_0 &= [B_0, B_1, B_2, \dots, B_r, (C_1 - A_1), (C_2 - A_2), \dots, (C_{r+1} - A_{r+1}), \\ &\quad - A_1, - A_2, \dots, - A_{r+1}] \end{aligned}$$

Recursive least squares algorithm for parameter update in the regression form

$$(y_{t+1} - \hat{y}_{ref_{t+1}}) + y_{ref_{t+1}} = \theta \phi_t \quad (ID)$$

is used to give parameter estimate $\hat{\theta}_{t+1}$.

The parameter estimate is used in turn to compute the control u_t in the following manner:

$$\text{Choose } u_t \text{ such that } \hat{\theta}_t \phi_t - y_{ref_{t+1}} = 0 \quad (\text{CONTROL})$$

Such a choice of u_t gives reference following in the minimum-variance sense.

ID equations are usually implemented as:

$$\hat{\theta}_t = \hat{\theta}_{t-1} + P_{t-1} \phi_{t-1} (\lambda + \phi_{t-1}^T P_{t-1} \phi_{t-1})^{-1} (y_t - \hat{\theta}_{t-1}^T \phi_{t-1})$$

$$P_t = \left[P_{t-1} - \frac{P_{t-1} \phi_{t-1} \phi_{t-1}^T P_{t-1}}{\lambda + \phi_{t-1}^T P_{t-1} \phi_{t-1}} \right] \frac{1}{\lambda}$$

Control u_t is computed by solving the linear set of equations

$$\hat{\theta}_t^T \phi_t - y_{\text{ref}_{t+1}} = 0.$$

Note that this involves inversion of a $p \times p$ matrix consisting of the first $p \times p$ matrix of $\hat{\theta}_t$.

This algorithm of (1) finding a linear least squares estimate in a linear regression model and (2) then computing the control to make the prediction based on the current parameter estimates equal to some known value, lies at the heart of a large number of "successful" adaptive control algorithms. The same basic algorithm is also used for recursive prediction algorithms.

Lattice Form Algorithm

Exact implementation of a recursive least-squares algorithm can be done using the joint-process ladder form with pre-windowing. The parameter estimate is in terms of the reflection coefficients and not in terms of $\hat{\theta}$. The reflection coefficients are used to compute the prediction and efficiently obtain the control u_t to make the predicted value equal to the desired value.

The proposed procedure involves $p+1$ iterations of computing prediction of $(y_{t+1} - y_{\text{ref}_{t+1}})$ for $u_t = 0, e_1, e_2, \dots, e_p, e_1$

denotes the i^{th} Euclidean basis vector $[0 \ 0 \ \dots \ 1 \ \dots \ 0]^T$ in E^p . Each such prediction involves $O(M)$ operations. The set of linear affine equations in u_t is then solved in $O(p^3)$ operations to obtain u_t . The total operation count thus remains $O(M)$, i.e., linear in the model order. Note that since we are using the ladder form merely to implement the exact equivalent of the algorithm using explicit estimation of $\hat{\theta}$, the convergence properties and stability property of the original algorithm continues to hold for the ladder case. Let us describe the algorithm in detail below:

(I) Initialize at time $t = 0$; $n = 0, \dots, M-1$

$$R_n^{-e}(0) = \delta I$$

$$R_n^{-r}(-1) = \delta I$$

$$F_n(0) = 0$$

$$r_n(0) = 0$$

$$\hat{y}_n(0) = 0$$

(II) At time t we have in memory $[R_n^{-e}(t-1)]$, $F_n(t-1)$, $F_n^y(t-1)$, $\hat{y}_n(t)$, $[R_n^{-r}(t-2)]$, $r_n(t-1)$, $K_n^y(t-1)$

Compute: $n = 0, \dots, M-1$

*(Remark on notation: The subscript n denotes the lattice stage. To avoid confusion, the time parameter t is now appearing in ().)

$$K_n(t-1) = F_n^T(t-1)[R_n^{-r}(t-2)]$$

$$K_n^*(t-1) = F_n(t-1)[R_n^{-e}(t-1)]$$

$$R_n^{-r}(t-1) = \left[R_n^{-r}(t-2) - \frac{R_n^{-r}(t-2)r_n(t-1)r_n^T(t-1)R_n^{-r}(t-2)}{\lambda(1-\beta_n(t)) + r_n^T(t-1)R_n^{-r}(t-2)r_n(t-1)} \right] \frac{1}{\lambda}$$

$$\beta_{n+1}(t) = \beta_n(t) + (1-\beta_n(t))^2 r_n^T(t-1)R_n^{-r}(t-1)r_n(t-1)$$

$$\beta_0(t) = 0$$

(III) Measurement $y(t)$ is used

$$\epsilon_n(t) = y(t) - \hat{y}_n(t) \quad n = 0, \dots, M-1$$

$$F_n^y(t) = \lambda F_n^y(t-1) + (1-\beta_n(t))r_n(t-1)\epsilon_n^T(t)$$

$$K_n^y(t) = F_n^y(t)[R_n^{-r}(t-1)]$$

(IV) Let ${}^i u(t) = e_i = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \quad i = 1, \dots, p$

Let ${}^0 u(t) = 0$

Let ${}^i x(t) = [{}^i u(t)^T, y^T(t), y_{\text{ref}}^T(t+1)]^T$

Compute for $i = 1, \dots, p; \quad n = 1, \dots, M$

$${}^i e_n(t) = {}^i e_{n-1}(t) + K_{n-1}(t-1) {}^i r_{n-1}(t-1)$$

$${}^i r_n(t) = {}^i r_{n-1}(t-1) + K_{n-1}^*(t-1) {}^i e_{n-1}(t)$$

$${}^i \hat{y}_n(t+1) = {}^i \hat{y}_{n-1}(t+1) - K_{n-1}^y(t) {}^i r_{n-1}(t)$$

$${}^i e_0(t) = {}^i r_0(t) = {}^i x(t)$$

giving p predictions ${}^i \hat{y}_M(t+1)$

(V) Solve the set of linear equations defined by

$$Au(t) = b$$

where

$$b = y_{\text{ref}}(t+1) - {}^0 \hat{y}_M(t+1)$$

and

$$\text{column } i \text{ of } A = ({}^i \hat{y}_n(t+1) - b)$$

This choice of $u(t)$ will give

$$\hat{y}_M(t+1) = y_{\text{ref}}(t+1) \text{ as desired.}$$

(VI) Apply control $u(t)$.

Use $x(t)$ formed by $u(t)$, $y(t)$, $y_{ref}(t)$ and compute for $n = 1, \dots, M$

$$e_0(t) = r_0(t) = x(t)$$

$$e_n(t) = e_{n-1}(t) + K_{n-1}(t-1)r_{n-1}(t-1)$$

$$r_n(t) = r_{n-1}(t-1) + K_{n-1}^*(t-1)e_{n-1}(t)$$

$$\hat{y}_n(t+1) = \hat{y}_{n-1}(t+1) - K_{n-1}^y(t)r_{n-1}(t)$$

$$(VII) \quad R_n^{-e}(t) = \left[R_n^{-e}(t-1) - \frac{R_n^{-e}(t-1)e_n(t)e_n^T(t)R_n^{-e}(t-1)}{\lambda(1-\beta_n(t)) + e_n^T(t)R_n^{-e}(t-1)e_n(t)} \right] \frac{1}{\lambda}$$

$$F_n(t) = \lambda F_n(t-1) + (1-\beta_n(t))r_n(t-1)e_n^T(t)$$

Go to step II